

Software –und Hardwareentwicklung eines Hobby-Fungariums

STUDIENARBEIT Studienmodul T3200

des Studienganges Mechatronik

an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Robin Michael Lägler und Sven Schelling

09. Mai 2021

Bearbeitungszeitraum

23. Oktober 2020 bis 09. Mai 2021

Matrikelnummer, Kurs

5112963 (R.L.) / 7458600 (S.S.), TMT18GR2

Ausbildungsfirma

Dürr Systems AG, 74321 Bietigheim

Betreuer der DHBW

Herr Tadeusz Rzedkowski

Erklärung der Eigenarbeit

Wir versichern hiermit, dass wir unsere Studien- und Projektarbeit mit dem Thema: „Software –und Hardwareentwicklung eines Hobby-Fungariums“, selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben.

Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Bietigheim-Bissingen, 09.05.2021

Ort

Datum

R. Lägge

Unterschrift

Bietigheim-Bissingen, 09.05.2021

Ort

Datum

J. Scheller

Unterschrift

Hinweis:

Der in dieser Dokumentation beschriebene Aufbau ist unabhängig von einer Implementierung als Fungarium oder Terrarium. In der vorliegenden Arbeit wird jedoch meist nur der Begriff Fungarium verwendet.

Aus Gründen der besseren Lesbarkeit wird bei Personenbezeichnungen und personenbezogenen Hauptwörtern in dieser Dokumentation die männliche Form verwendet. Entsprechende Begriffe gelten im Sinne der Gleichbehandlung grundsätzlich für alle Geschlechter. Die verkürzte Sprachform hat nur redaktionelle Gründe und beinhaltet keine Wertung.

Kurzfassung

Kein Lebewesen der Welt hat im Laufe der Geschichte so unterschiedliche Interpretationen erhalten wie der Pilz. Die verschiedenen Pilzarten unterscheiden sich dabei vor allem aufgrund Ihrer unterschiedlichen Lebensräume.

Eine besondere Gruppe von Pilzen bilden die sogenannten Speisepilze. Unter dem Begriff Speisepilze werden die Pilzarten verstanden, welche wohlschmeckend und nicht schädlich für den menschlichen Körper sind.

Da der Speisepilz mittlerweile zu den beliebtesten Nahrungsmitteln gehört, wird vermehrt auf eine private Pilzzucht gesetzt. Speisepilze haben dabei sehr spezifische Anforderungen an ihre Umweltbedingungen, weshalb eine genaue Überwachung erforderlich ist. Gute Wachstumsbedingungen für Pilze sind dabei meist unbeheizte Räume mit Temperaturen zwischen 10 - 20°C. Des Weiteren benötigen eine Vielzahl von Pilzen nur wenig Licht. Für eine gelungene Pilzzucht ist dagegen meist eine hohe Luftfeuchtigkeit als auch eine ausreichende Belüftung notwendig, um das Austrocknen der Pilzzuchtkulturen zu vermeiden.

Um den verschiedenen Anforderungen auch in normal klimatisierten Räumlichkeiten gerecht zu werden, beschäftigt sich diese Dokumentation mit dem Aufbau eines privaten Fungariums. Dabei sollen die verschiedenen Parameter wie notwendige Luftfeuchtigkeit, Lichtdauer oder auch die Temperatur über eine individuell erstellte Steuer Webseite einstellbar sein. Die erforderlichen Parametergrenzen können dabei je Pilzbrut stark voneinander abweichen und können über die Steuer Webseite der jeweiligen Pilzbrut angepasst werden.

Die Technik basiert dabei auf einer Reihe von Technologien, mit denen die Prinzipien der Automatisierung angewendet werden. Dadurch soll zum einen die Ausbeute gesteigert, als auch die manuelle Verpflegung auf ein Minimum reduziert werden. Die folgende Dokumentation beschreibt schrittweise die Erstellung eines automatisierten Fungariums mit den grundlegenden Komponenten.

Zusätzlich ist festzuhalten, dass diese Dokumentation lediglich eine Möglichkeit zum Aufbau eines Hobby-Fungariums ist. Dabei soll in erster Linie die Anwendung von Automatisierungstechniken demonstriert werden, welche wiederum einfach an andere Systeme, wie beispielsweise einem Terrarium, angepasst werden können.

Inhaltsverzeichnis

Kurzfassung.....	III
I. Abkürzungsverzeichnis	VII
II. Tabellenverzeichnis	VIII
III. Abbildungsverzeichnis	X
1 Einleitung	1
2 Überblick	3
2.1 Software	4
2.1.1 Steuerseite	4
2.1.2 Steuerungsseite Use Case Diagramme	13
2.1.3 Steuerprogramm	15
2.1.4 Steuerprogramm Use Case Diagramm	22
2.2 Hardware	24
3 Port Weiterleitung.....	25
3.1 Portfreigabe	26
4 Domain und DynDNS.....	27
4.1 Domain und DynDNS über DDNSS.....	28
5 Installation über Bash-Skript	29
6 LAMP	31
6.1 Linux und Raspbian	31
6.2 Apache Server	32
6.2.1 Allgemeines.....	32
6.2.2 Installation	33
6.3 Virtual Host	34
6.4 Erstellung Virtual Host	34
6.5 .htaccess und .htpasswd	37
6.5.1 Einbinden der „.htaccess“- Datei	37
6.6 HTTPS und SSL	40

6.6.1 Installation ACME Certbot.....	41
7 PHP.....	42
7.1 Installation	42
8 Datenbanksystem	43
8.1 MySQL/MariaDB.....	44
8.1.1 Allgemeines zu MySQL/MariaDB	44
8.1.2 Funktionsprinzip MySQL/MariaDB	45
8.1.3 Installation MariaDB	45
8.2 PHPMyAdmin	46
8.2.1 Installation PHPMyAdmin.....	46
8.3 Verwendung im Projekt.....	48
8.4 Datenbank erstellen.....	48
8.5 Datenbanktabellen.....	49
8.5.1 Tabelle Einstellungen.....	49
8.5.2 Tabelle Protokoll	50
8.5.3 Tabelle DHT	51
8.5.4 Tabelle Manuell.....	51
8.5.5 Tabelle Voreinstellungen.....	52
9 PHP- Steuerungsseite.....	53
9.1 Programmiersprachen	53
9.2 Struktur des Programmcodes	54
10 Java-Steuerprogramm	57
10.1 Installation	57
10.2 Struktur des Programmcodes	58
10.3 Logger (log4j).....	62
10.4 Update gpio	63
10.5 Auslesen der Messwerte des DHT22 – Sensor	63
10.5.1 Adafruit.....	64

10.6 Herunterladen der Java-Applikation.....	65
10.7 Einrichten eines Autostarts	67
11 Hardware	68
11.1 Fungarium/Terrarium	68
11.2 Lüftungssystem.....	69
11.3 Licht 72	
11.4 DHT22-Sensor.....	73
11.5 Verteilerbox mit Raspberry Pi	73
11.5.1 Seitenstabilisierung	74
11.5.2 Elektrische Verkabelung.....	75
11.5.3 Schutzbdeckung	77
11.5.4 Relaisboard und Raspberry Pi.....	78
11.6 Zusammenbau.....	79
11.7 Weitere Komponenten	80
12 Fazit und Ausblick	81
IV. Literaturverzeichnis	XII
13 Anhang.....	XIV
13.1 Dokumentation der Steuerseite Use Cases.....	XIV
13.2 Dokumentation der Steuerprogramm Use Cases	XXXI
13.3 Komponentenliste	L1
13.4 Fertigungszeichnungen	LII
13.5 Elektrischer Schaltplan	LXVI

I. Abkürzungsverzeichnis

ASF	Apache Software Foundation
CSS	Cascading Style Sheets
DB	Datenbank
DBMS	Datenbank-Managementsystem
DHT-Sensor	digitaler Temperatur -und Feuchtigkeitssensor
GCT	Global Control Table
GPIO	General Purpose Input-Output
HTTP	Hypertext Transport Protocol
HTTPS	Hypertext Transport Protocol Secure
HTML	Hypertext Markup Language
IDE	integrierte Programmierumbgebung
JAR	Java Archive
JS	JavaScript
PHP	Hypertext Preprocessor
SchuKo	Schutz- Kontakt
SSL	Secure Socket Layers
TLS	Transport Layer Security
URL	Uniform Resource Locators
WPI	WiringPi

II. Tabellenverzeichnis

Tabelle 1: Parameterbedingungen	15
Tabelle 2: Default-Werte	16
Tabelle 3: Hardwarekomponentenliste	24
Tabelle 4: Automatische Einstellungen übertragen	XIV
Tabelle 5: Manuelles Starten vorhandener Aktoren	XV
Tabelle 6: Starten manueller Aktionen	XVI
Tabelle 7: Auswahl des Steuermodi	XVII
Tabelle 8: Voreinstellung anzeigen	XVIII
Tabelle 9: Auswahl des Verwendungszwecks	XIX
Tabelle 10: Auswahl der Lichtfunktion	XX
Tabelle 11: Auswahl der vorhandenen Komponenten	XXI
Tabelle 12: Pinbelegung einsehen	XXII
Tabelle 13: Auswahl der Pinbelegung	XXIII
Tabelle 14: Temperaturverlauf in Diagramm anzeigen	XXIV
Tabelle 15: Temperaturverlauf in Tabelle anzeigen	XXV
Tabelle 16: Feuchtigkeitsverlauf in Diagramm anzeigen	XXVI
Tabelle 17: Feuchtigkeitsverlauf in Tabelle anzeigen	XXVII
Tabelle 18: Protokoll in Tabelle anzeigen	XXVIII
Tabelle 19: Nach Protokolldaten suchen	XXIX
Tabelle 20: Zustand der Komponenten anzeigen	XXX
Tabelle 21: Verbindung zur Datenbank aufbauen	XXXI
Tabelle 22: Voreinstellungen Einlesen	XXXII
Tabelle 23: Systemkonfiguration setzen	XXXIII
Tabelle 24: Datenbank "Protokoll" aktualisieren	XXXIV
Tabelle 25: Datenbank "manuell" aktualisieren	XXXV
Tabelle 26: Sensordaten einlesen	XXXVI
Tabelle 27: Sensordaten weiterverarbeiten	XXXVII
Tabelle 28: Sensorneustart	XXXVIII
Tabelle 29: Befeuchtungsroutine starten	XXXIX
Tabelle 30: Fogger steuern	XL
Tabelle 31: Lüftung niedertourig steuern	XLI
Tabelle 32: Lüftung starten	XLII
Tabelle 33: Lüftung hochtourig steuern	XLIII

Tabelle 34: Licht steuern	XLIV
Tabelle 35: Heizung steuern.....	XLV
Tabelle 36: Kühlung steuern.....	XLVI
Tabelle 37: Protokollieren in Datei.....	XLVII
Tabelle 38: Wechsel der Betriebsart.....	XLVIII
Tabelle 39: Wechsel von Normalbetrieb in Notbetrieb.....	XLIX
Tabelle 40: Wechsel von Notbetrieb in Normalbetrieb.....	XLIX
Tabelle 41: Einstellparameter einlesen.....	L

III. Abbildungsverzeichnis

Abbildung 1: Gesamtüberblick.....	3
Abbildung 2: Menü Seite automatischer Betrieb.....	5
Abbildung 3: Pop-Up-Fenster Übermittelte Daten	6
Abbildung 4: Menü Seite manueller Betrieb.....	7
Abbildung 5: Voreinstellungen	8
Abbildung 6: Header Terrarium	8
Abbildung 7: GPIO-Pinbelegung.....	9
Abbildung 8: Temperatur Seite	10
Abbildung 9: Luftfeuchtigkeit Seite	11
Abbildung 10: Protokoll Seite.....	12
Abbildung 11: Use Case Diagramm Menüseite	13
Abbildung 12: Use Case Diagramm Temperatur-, Feuchtigkeits -und Protokollseite	14
Abbildung 13: Use Cases Steuerprogramm	23
Abbildung 14: Port Weiterleitung	25
Abbildung 15: Freigaben Fritzbox.....	26
Abbildung 16: Geräteauswahl Portfreigabe	26
Abbildung 17: Portfreigaben aktiviert.....	26
Abbildung 18: DDNSS Domain erstellen	28
Abbildung 19: Konfiguration Domain	28
Abbildung 20: Anmeldedaten DynDNS-Anbieter	28
Abbildung 21: Schaubild Serveraufruf	32
Abbildung 22: index.html Datei	33
Abbildung 23: Steuerungsseite Menü.....	36
Abbildung 24: Schaubild SSL	40
Abbildung 25: Schaubild Datenbanksystem	43
Abbildung 26: Beispieldatenebene RDBMS.....	44
Abbildung 27: Schaubild Funktionsprinzip MySQL	45
Abbildung 28: Schaubild phpMyAdmin	46
Abbildung 29: PHPMyAdmin-Oberfläche.....	47
Abbildung 30: Schaubild PHP/ MariaDB/ Java	48
Abbildung 31: Tabellen in DB	49
Abbildung 32: Tabelle einstellungen.....	49
Abbildung 33: Tabelle protokoll	50

Abbildung 34: Tabelle dht	51
Abbildung 35: Tabelle manuell	51
Abbildung 36: Tabelle Voreinstellungen	52
Abbildung 37: Schaubild Programmstruktur PHP	54
Abbildung 38: Auswahl Java-Version	58
Abbildung 39: Programmstruktur Java	58
Abbildung 40: Struktur Fungarium Ordner	65
Abbildung 41: Aquarium	68
Abbildung 42: Lüfter	69
Abbildung 43: Fogger	69
Abbildung 44: Lüftungssystem.....	69
Abbildung 45: Fogger im Eimer	70
Abbildung 46: Befestigung Lüfter.....	70
Abbildung 47: Rohrvorrichtung	71
Abbildung 48: Rohrwanne	71
Abbildung 49: Schlauch zum Abtransport.....	71
Abbildung 50: Ansteuerung Lüfter	72
Abbildung 51: Licht	72
Abbildung 52: DHT22-Sensor.....	73
Abbildung 53: Verteilerbox.....	74
Abbildung 54: Seitenstreben.....	74
Abbildung 55: Hauptstecker/Zuleitung.....	75
Abbildung 56: Innenleben Verteilerbox	76
Abbildung 57: Schutzabdeckung	77
Abbildung 58: Schutzabdeckung der Steckdosenkontakte	77
Abbildung 59: Relaisboard und Raspberry Pi	78
Abbildung 60: Verkabelung Steuerteil	78
Abbildung 61: Zusammenbau Fungarium 1	79
Abbildung 62: Zusammenbau Fungarium 2	79

1 Einleitung

Das zentrale Herzstück des automatisierten Hobby–Fungariums ist ein Raspberry Pi Single Board Computer, welcher von der britischen Raspberry Pi Foundation entwickelt wurde. Es handelt sich dabei um einen sehr einfachen Kleinrechner, der ungefähr die Größe einer Kreditkarte einnimmt. Ursprünglich war die Idee hinter der Entwicklung des Raspberry Pi jungen Leuten das Erproben von Programmiersprachen zu erleichtern. Inzwischen kommt der Einplatinencomputer hauptsächlich bei Kleinprojekten als zentrale Steuereinheit zum Einsatz. Aufgrund des sehr einfach gehaltenen Aufbaus ist der Raspberry Pi günstig zu erwerben. Dennoch verfügt der Kleinrechner über alle Komponenten, die für den Aufbau eines Hobby-Fungariums notwendig sind. Die wichtigste Eigenschaft des Raspberry Pi für die in dieser Dokumentation erarbeitete Anwendung sind seine GPIO-Pins (General Purpose Input-Output). Diese Ein- und Ausgänge werden im Folgenden primär zur Ansteuerung der Relais und zum Messen von Sensordaten verwendet. Durch die Ansteuerung der verschiedenen Hardwarekomponenten werden die Umweltbedingungen gezielt manipuliert, um den Anforderungen der eingesetzten Pilzbrut gerecht zu werden.

Die Software, welche auf dem Raspberry läuft, übernimmt dabei die Aufgabe der Automatisierung. Insgesamt ist der Softwareteil in zwei Bereiche zu unterteilen:

- PHP → Konfigurationswebseite
- Java → Steuersoftware

Auf beide Bereiche wird im Laufe der Dokumentation genauer eingegangen.

Die Dokumentation ist in vier Abschnitte unterteilt. Im ersten Teil wird ein Überblick über die verschiedenen Use Cases, Funktionen der Soft -und Hardware und der Verwendung der einzelnen Hardwarekomponenten gegeben.

Darauf folgt ein großer Abschnitt, in welchem die Einrichtung und Installation des Apache Webservers thematisiert wird. In diesem Teil der Dokumentation wird ebenfalls auf die Erstellung einer Domain und die dafür benötigte Konfiguration des Routers eingegangen. Des Weiteren wird die Einrichtung einer Datenbank behandelt, um die PHP-Konfigurationsseite mit dem Java-Steuerungsprogramm zu verbinden. Anschließend folgt ein Abschnitt, in welchem der Aufbau des Programmcodes der PHP-Seite und des Java-Programms kurz erläutert werden.

Zum Schluss der Dokumentation, wird der Aufbau der Hardware beleuchtet. Darin wird das Projektergebnis visuell dargestellt.

Allgemein ausgedrückt stellt diese Dokumentation mehr ein Handbuch oder eine Installationsanleitung für den Entwickler eines Hobby-Fungariums dar. Das Augenmerk liegt dabei mehr auf der Installation, Konfiguration und Einrichtung des Raspberry Pi, als auf der detaillierten Beschreibung des entwickelten Programmcodes. Das Ziel der Dokumentation ist, dass jeder Hobbyentwickler sich ein teilautomatisiertes Fungarium oder Terrarium mit einer Schritt-für-Schritt-Anleitung selbst aufbauen kann.

2 Überblick

In diesem Abschnitt sollen die im Hobby-Fungarium umgesetzten Funktionen beschrieben werden. Um den folgenden Zusammenhang einfacher verstehen zu können, ist in Abbildung 1 ein Gesamtüberblick dargestellt. In der Darstellung sind zum einen die verschiedenen Komponenten als auch die Verbindungen zwischen den Komponenten abgebildet. Zusätzlich werden über die rot umrahmten Kästchen die jeweilige Aktion der Verbindung beschrieben.

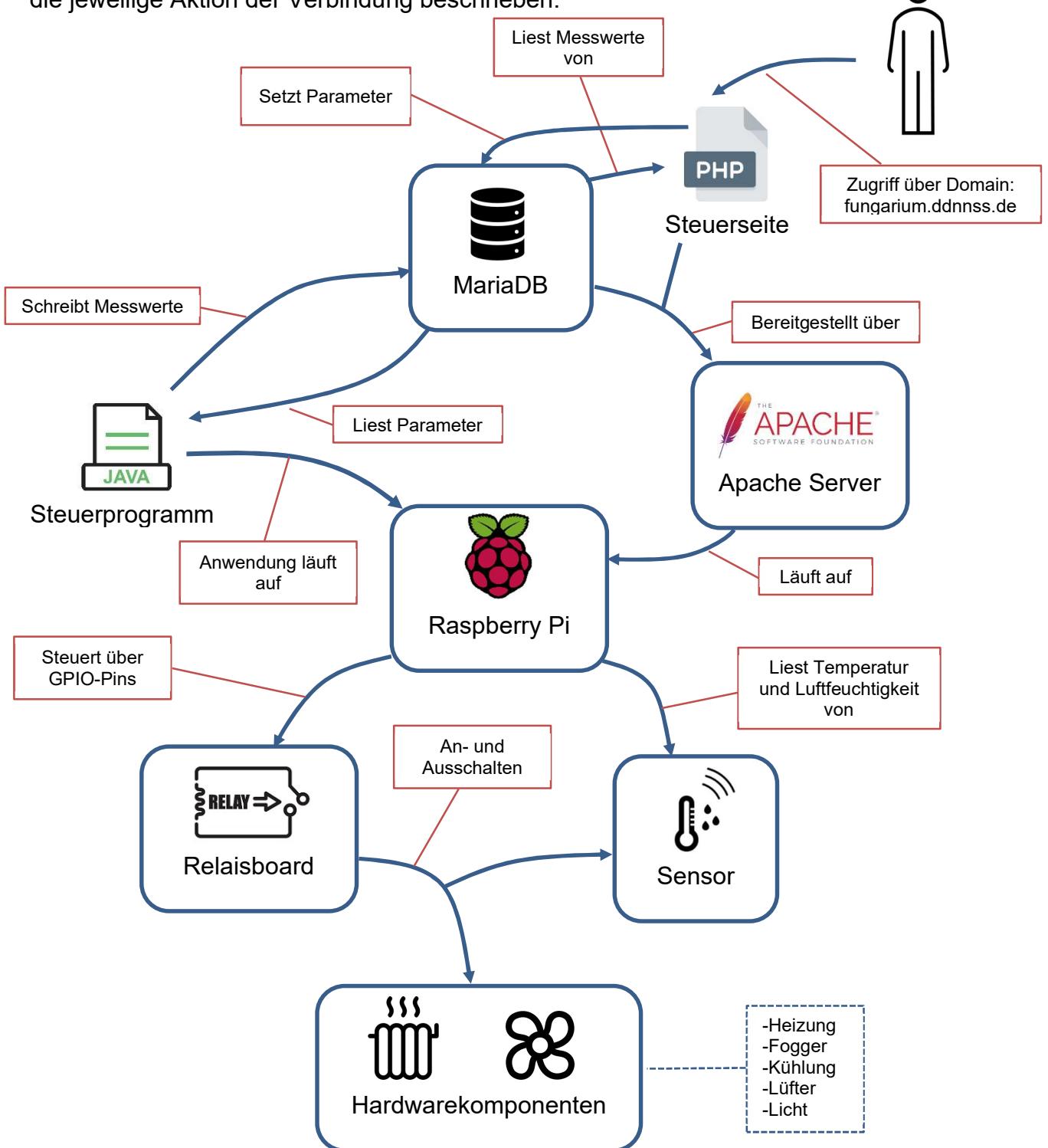


Abbildung 1: Gesamtüberblick

2.1 Software

Der Raspberry Pi läuft auf dem Betriebssystem Raspbian. Die zentrale Steuerungssoftware ist eine Java-Applikation, welche die aktuell anzuwendenden Parameter der Datenbank entnimmt. Zusätzlich liest das Java-Programm die aktuelle Luftfeuchtigkeit und Temperatur in regelmäßigen, vorgegebenen Zeitabständen, über den DHT-Sensor aus und schreibt diese zusammen mit dem Protokoll in die Datenbank.

Des Weiteren läuft auf dem Kleinrechner ein Apache-Webserver. Auf diesem Server wird eine in PHP, HTML, JS und CSS verfasste Steuerwebseite bereitgestellt, über die der Benutzer die Parameter des Steuerprogramms einstellen kann. Zusätzlich werden auf dieser Seite die aktuellen Messwerte tabellarisch sowie grafisch dargestellt.

Der Zugang zur Steuerseite ist dabei über eine „htaccess“- und „htpasswd“-Datei abgesichert. Des Weiteren wird die Server-Client-Verbindung mit Hilfe der Einbindung eines SSL-Zertifikats gesichert.

2.1.1 Steuerseite

Die Steuerseite ist unter der erstellten Domain erreichbar: „example.ddnss.de“. Nach Eingabe und Bestätigung der Domain muss der erforderliche Benutzername und das dazugehörige Passwort eingegeben werden. Ist die Eingabe erfolgreich, öffnet sich die Steuerseite. Sowohl die Erstellung der Domain als auch die Absicherung des Zugangs sind im Laufe dieser Dokumentation ausführlich beschrieben.

Die Steuerseite umfasst insgesamt fünf Reiter:

- Menü
- Temperatur
- Luftfeuchtigkeit
- Protokoll

Zwischen diesen Reitern kann über ein Dropdownmenü auf jeder Seite hin und her gewechselt werden.

Menü:

Menü ist der erste Reiter und wird deshalb beim Aufruf der Steuerseite über die entsprechende Domain standardmäßig zuerst angezeigt. Im Menü können die verschiedenen Einstellparameter vom Anwender angepasst werden. Zu den einstellbaren Parametern zählen beispielsweise der Lichtstart und das Lichtende, die Maximal- und Mindesttemperatur, die Mindestfeuchtigkeit und die Befeuchtungs- und Lüftungsdauer. Bei Eingabe eines Parameters außerhalb des zulässigen Bereichs, wird das Eingabefeld mit einem roten Rahmen markiert. Ein Ausschnitt der Einstellungsseite ist in Abbildung 2 dargestellt.

FUNGARIUM

Licht Einstellungen

Lichtstart:
16:02 Daten absenden

Lichtende:
16:06 Daten absenden

Temperatur Einstellungen

Mindesttemperatur:
22 °C Daten absenden

Höchsttemperatur:
33 °C Daten absenden

Mindesttemperatur nachts:
23 °C Daten absenden

Höchsttemperatur nachts:
33 °C Daten absenden

Luftfeuchtigkeit Einstellungen

Mindestfeuchtigkeit:
99 % Daten absenden

Befeuchtungsdauer:
5 min Daten absenden

Lüftung Einstellungen

Lüftungsdauer:
6 min Daten absenden

Lüftungsintervall:
255 min Daten absenden

Steuerungsauswahl

Auto/Manuell:

Voreinstellungen:

Abbildung 2: Menü Seite automatischer Betrieb

Änderungen an den Parametern müssen über einen „Submit“ Button bestätigt werden. Mit der Bestätigung des Buttons werden die aktualisierten Daten an die Datenbank gesendet. Bevor dies geschieht, erscheint ein Pop-Up-Fenster mit den zu übermittelnden Daten. Dieses Pop-Up-Fenster ist mit dem Modul „Sweetalarm.js“ generiert und ist in Abbildung 3 dargestellt.

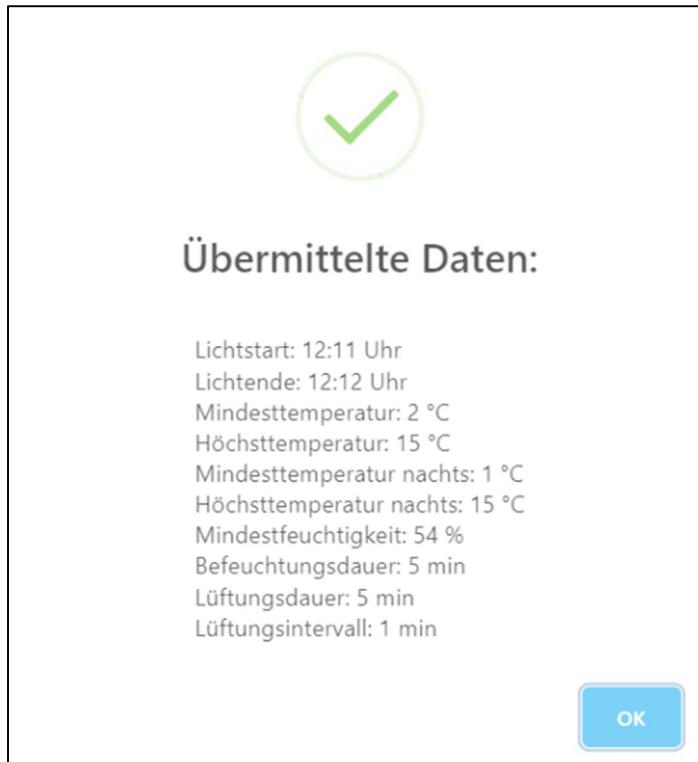


Abbildung 3: Pop-Up-Fenster Übermittelte Daten

Nach der Betätigung des „Ok“-Buttons vom Anwender werden die Daten an die Datenbank übermittelt.

Mit der Checkbox „Auto/Manuell“ kann vom automatischen in den manuellen Betrieb des Systems gewechselt werden. Wenn der Schalter auf „Auto“ steht, ist das System im automatischen Betrieb und die Menü-Seite zeigt die automatischen Einstellungen, die vom Anwender ausgewählt werden können. Ist die Checkbox aktiviert, wird das System in den manuellen Modus versetzt und die automatischen Einstellungen werden auf der Steuerungsseite ausgeblendet. Gleichzeitig werden die manuellen Einstellungen sichtbar. Die Menü-Seite im manuellen Modus ist in Abbildung 4 dargestellt.

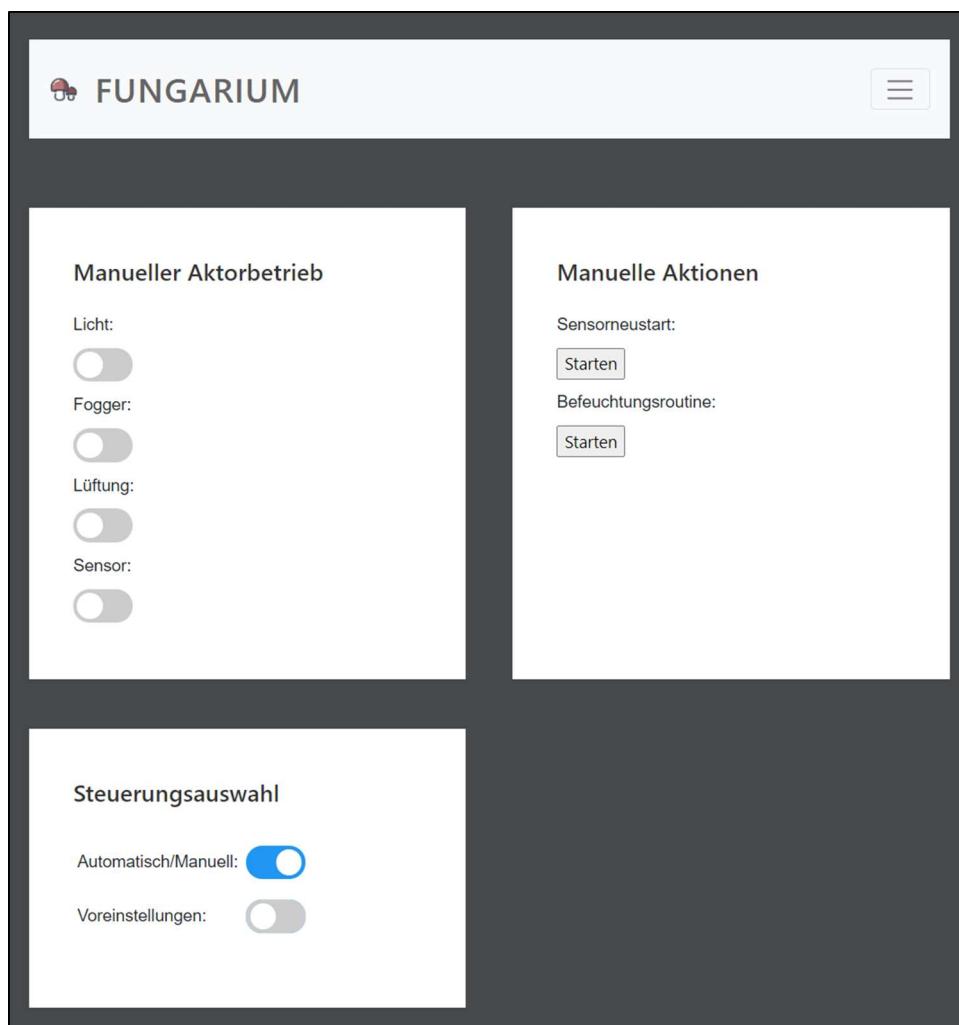


Abbildung 4: Menü Seite manueller Betrieb

Dabei können im Fenster „Manueller Aktorbetrieb“ die ausgewählten Komponenten, welche in den Voreinstellungen aktiviert sind, manuell ein -oder ausgeschalten werden. Die aktuellen Zustände der Komponenten werden in einer Datenbank gespeichert. In „Manuelle Aktionen“ kann vom Anwender ein Sensorneustart oder eine Befeuchtungsroutine angestoßen werden.

Mit der Checkbox „Voreinstellungen“, die standartmäßig ausgeschalten ist, können die Voreinstellungen des Systems vorgenommen werden (siehe Abbildung 5).

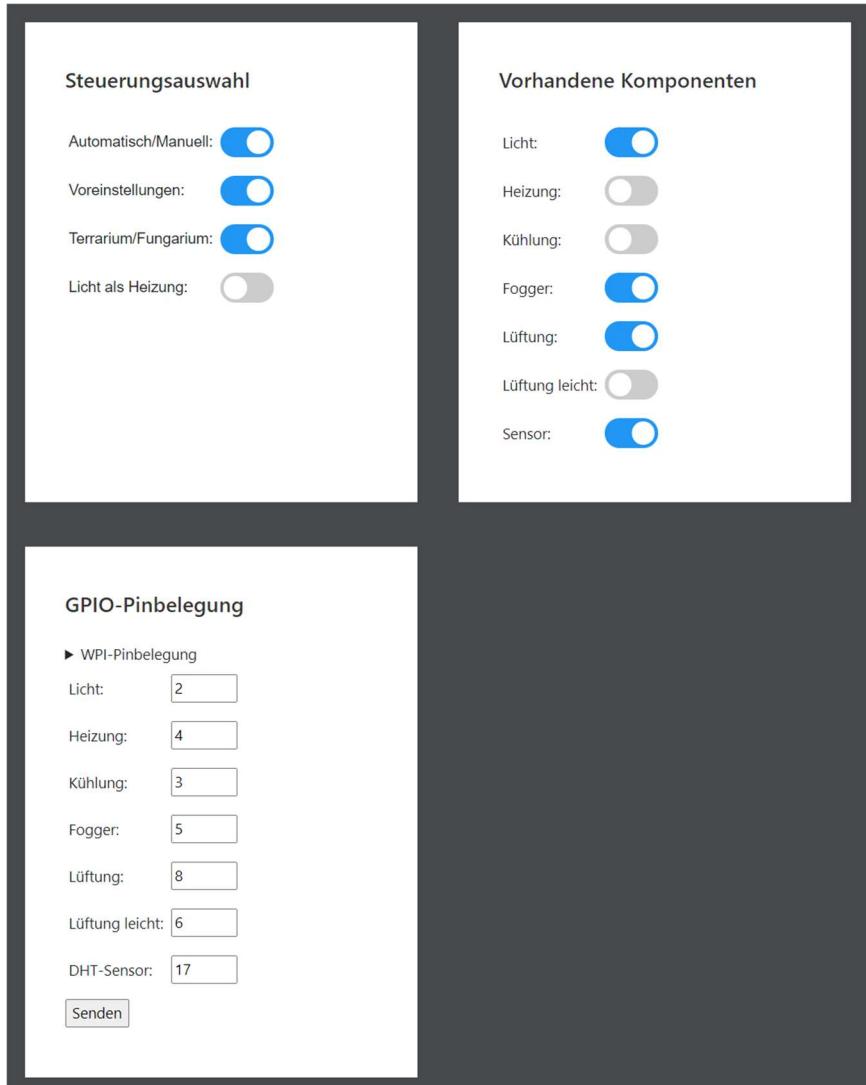


Abbildung 5: Voreinstellungen

Dabei wird auf der einen Seite das Fenster „Steuerungsauswahl“ mit den Checkboxen „Terrarium/Fungarium“ und „Licht als Heizung“ erweitert. Mit der Checkbox „Terrarium/Fungarium“ kann die jeweilige Verwendung des Projekts ausgewählt werden. Mit dieser Checkbox werden die Überschrift und das Icon der Webseite an den Verwendungszweck des Anwenders angepasst. Wird die Steuerungsseite für ein Terrarium verwendet, kann die Optik der „Headers“ durch diese Checkbox angepasst werden (siehe Abbildung 6).

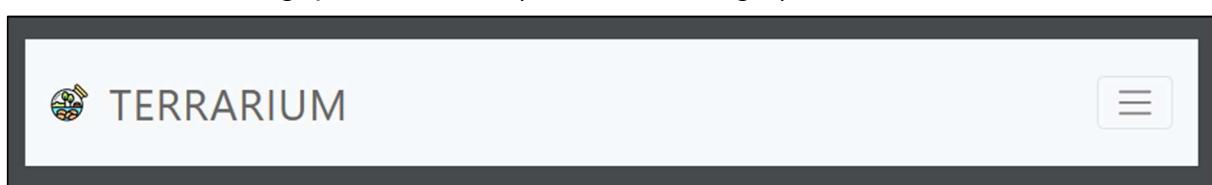


Abbildung 6: Header Terrarium

Mit der Checkbox „Licht als Heizung“ kann das Licht bei fehlender Heizung zusätzlich zur Erhöhung der Temperatur verwendet werden. Dies ist eine Voreinstellung, die sich auf den automatischen Betrieb beschränkt. Die Einstellung ist speziell im Bereich der Terrarium Automatisierung hilfreich.

Auf der anderen Seite wird das Fenster „Vorhandene Komponenten“ sichtbar, in welchem der Anwender seine im Projekt verwendeten Bauteile zur Steuerung des Fungarium -oder Terrariums konfigurieren kann. Mit dieser Einstellung werden die verfügbaren Komponenten ein -oder ausgeblendet, die im Fenster „manueller Aktorbetrieb“ gesteuert werden können. Ebenfalls werden anhand der Einstellungen auf der Seite „Protokoll“ die ausgewählten Zustandsdiagramme der Komponenten erstellt.

Des Weiteren wird der Block „GPIO-Pinbelegung“ sichtbar, in welchem die WPI-Pinbelegung eines Raspberry Pi 3 über ein Dropdown-Menü angezeigt werden kann (siehe Abbildung 7).

WPI (WiringPi) ist eine Pin-basierte GPIO Bibliothek, die für viele Anwendungen, wie bspw. der Integrierung eines DHT-Sensors auf dem Raspberry Pi benötigt wird. Sie beschreibt die Pinbelegung auf dem Raspberry Pi.

Des Weiteren können die jeweiligen Pins der Komponenten an die Datenbank übergeben werden, sodass die Ansteuerung der korrekten Relais und Komponenten durch das Raspberry Pi sichergestellt wird. Bevor die Daten übertragen werden, wird analog zu den Einstellungen ein Pop-Up-Fenster zur Übersicht der Pins angezeigt.

GPIO-Pinbelegung

▼ WPI-Pinbelegung

GPIO#	NAME		NAME	GPIO#
	3.3 VDC Power		5.0 VDC Power	
8	GPIO 8 SDA1 (I2C)	1	5.0 VDC Power	
9	GPIO 9 SCL1 (I2C)	2	Ground	
7	GPIO 7 GPCLK0	3	GPIO 15 TxD (UART)	15
	Ground	4	GPIO 16 RxD (UART)	16
0	GPIO 0	5	GPIO 1 PCM_CLK/PWM0	1
2	GPIO 2	6	Ground	
3	GPIO 3	7	GPIO 4	4
	3.3 VDC Power	8	GPIO 5	5
12	GPIO 12 MOSI (SPI)	9	Ground	
13	GPIO 13 MISO (SPI)	10	GPIO 6	6
14	GPIO 14 SCLK (SPI)	11	GPIO 10 CE0 (SPI)	10
	Ground	12	GPIO 11 CE1 (SPI)	11
30	SDAO (I2C ID EEPROM)	13	SCL0 (I2C ID EEPROM)	31
21	GPIO 21 GPCLK1	14	Ground	
22	GPIO 22 GPCLK2	15	GPIO 26 PWM0	26
23	GPIO 23 PWM1	16	Ground	
24	GPIO 24 PCM_FS/PWM1	17	GPIO 27	27
25	GPIO 25	18	GPIO 28 PCM_DIN	28
	Ground	19	GPIO 29 PCM_DOUT	29

Verwenden folgender Pinbelegung: **GPIO#**

Abbildung 7: GPIO-Pinbelegung

Temperatur:

Unter dem Reiter Temperatur befindet sich sowohl eine graphische als auch eine tabellarische Darstellung der gemessenen DHT-Sensordaten (siehe Abbildung 8). Die Genauigkeit des Diagrammbereichs kann dabei über einfaches zoomen beliebig verändert werden.

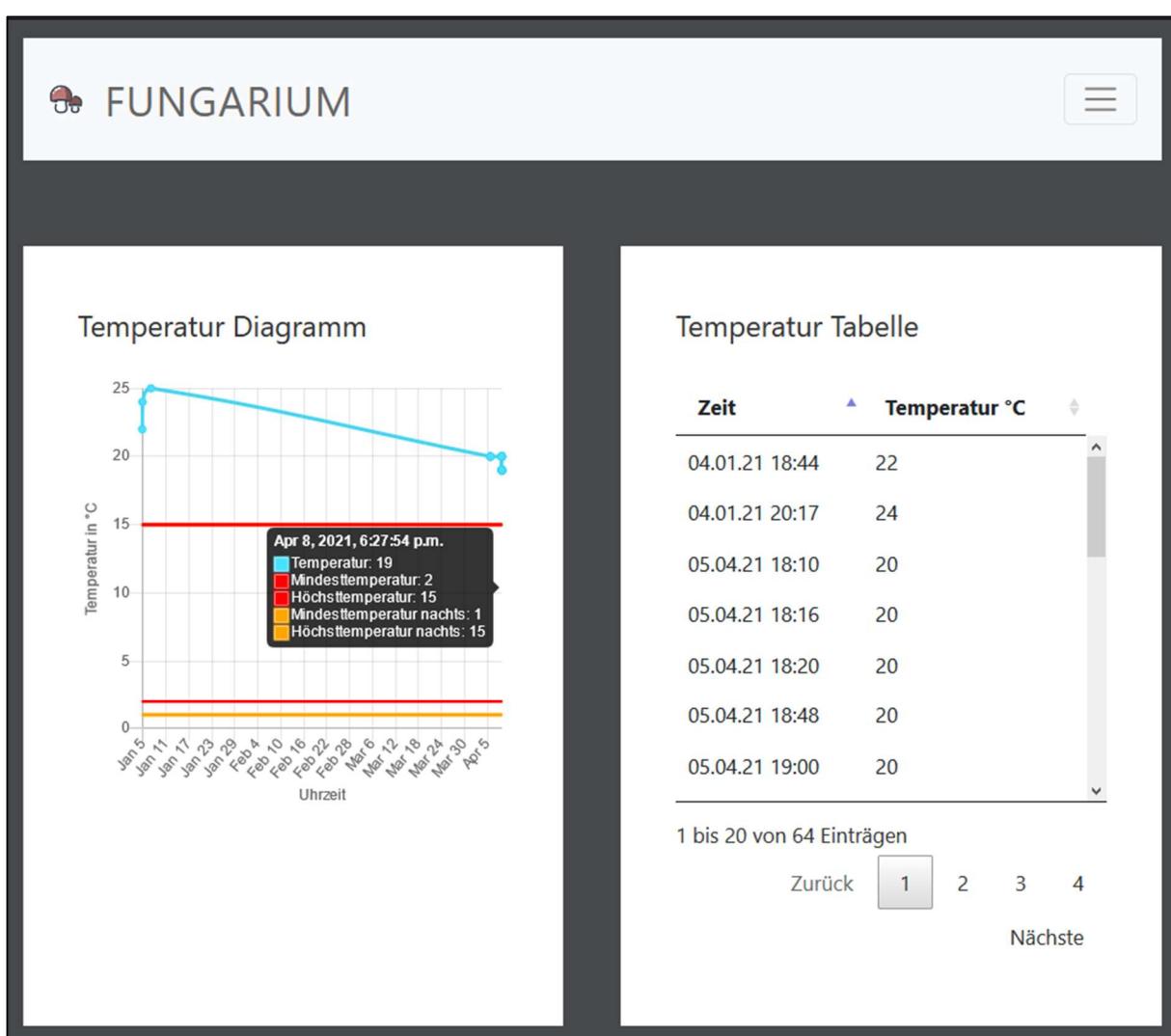


Abbildung 8: Temperatur Seite

Die eingestellte Höchst- und Mindesttemperatur tagsüber wird mit roten, horizontalen Linien dargestellt. Die Höchst- und Mindesttemperatur, die für die Nacht eingestellt sind, werden orange dargestellt.

Luftfeuchtigkeit:

Die Luftfeuchtigkeitsseite ist in ihrer Struktur analog zur Temperaturseite aufgebaut (siehe Abbildung 9):

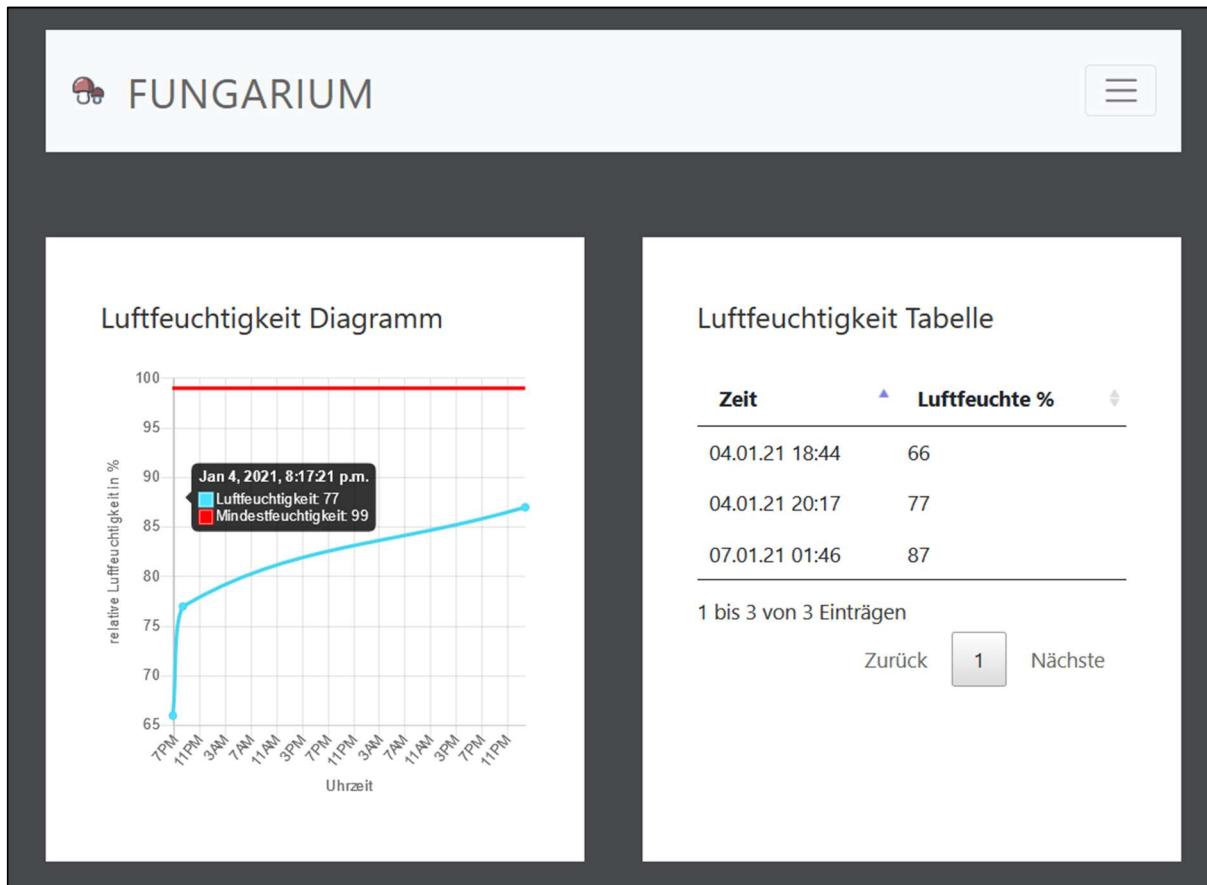


Abbildung 9: Luftfeuchtigkeit Seite

Die eingestellte Mindestfeuchte wird mit einem roten horizontalen Graphen dargestellt.

Protokoll:

Der Reiter Protokoll dient zur Benachrichtigung des Anwenders bei möglichen Fehlermeldungen (z.B. Fehlerhafte Messung) oder Zustandsänderungen der Hardwarekomponenten (z.B. Heizung eingeschalten). Zusätzlich ist unter diesem Reiter ein Zustands-Zeit-Diagramm von den jeweiligen Hardwarekomponenten, welche zur Manipulation der Umweltbedingungen eingesetzt werden, zu finden. In den Zustands-Zeit-Diagrammen kann abgelesen werden, zu welchen Zeitpunkten die entsprechenden Hardwarekomponenten ein- bzw. ausgeschalten wurden. Der Wert „1“ steht dabei für den Zustand „An“ und „0“ für den Zustand „Aus“. Ein Ausschnitt der Protokollseite ist in Abbildung 10 dargestellt.

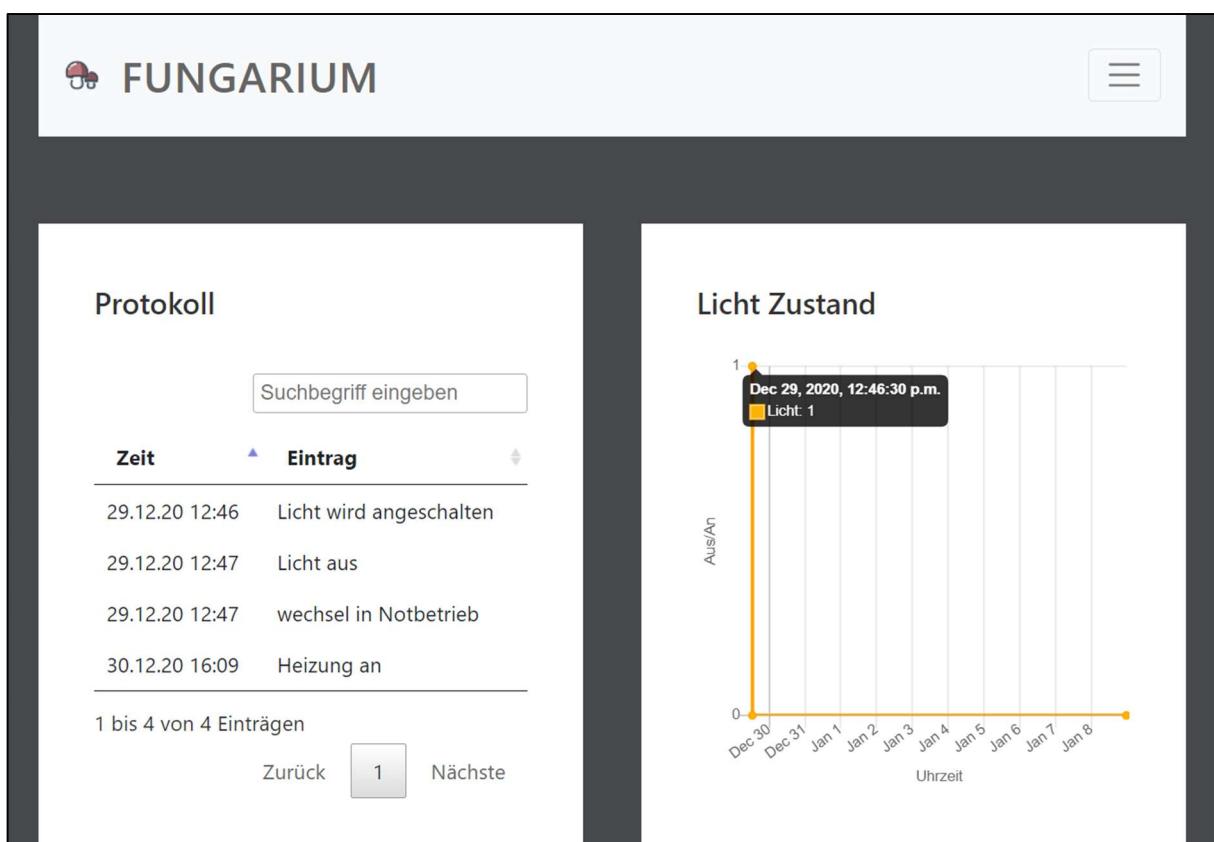


Abbildung 10: Protokoll Seite

2.1.2 Steuerungsseite Use Case Diagramme

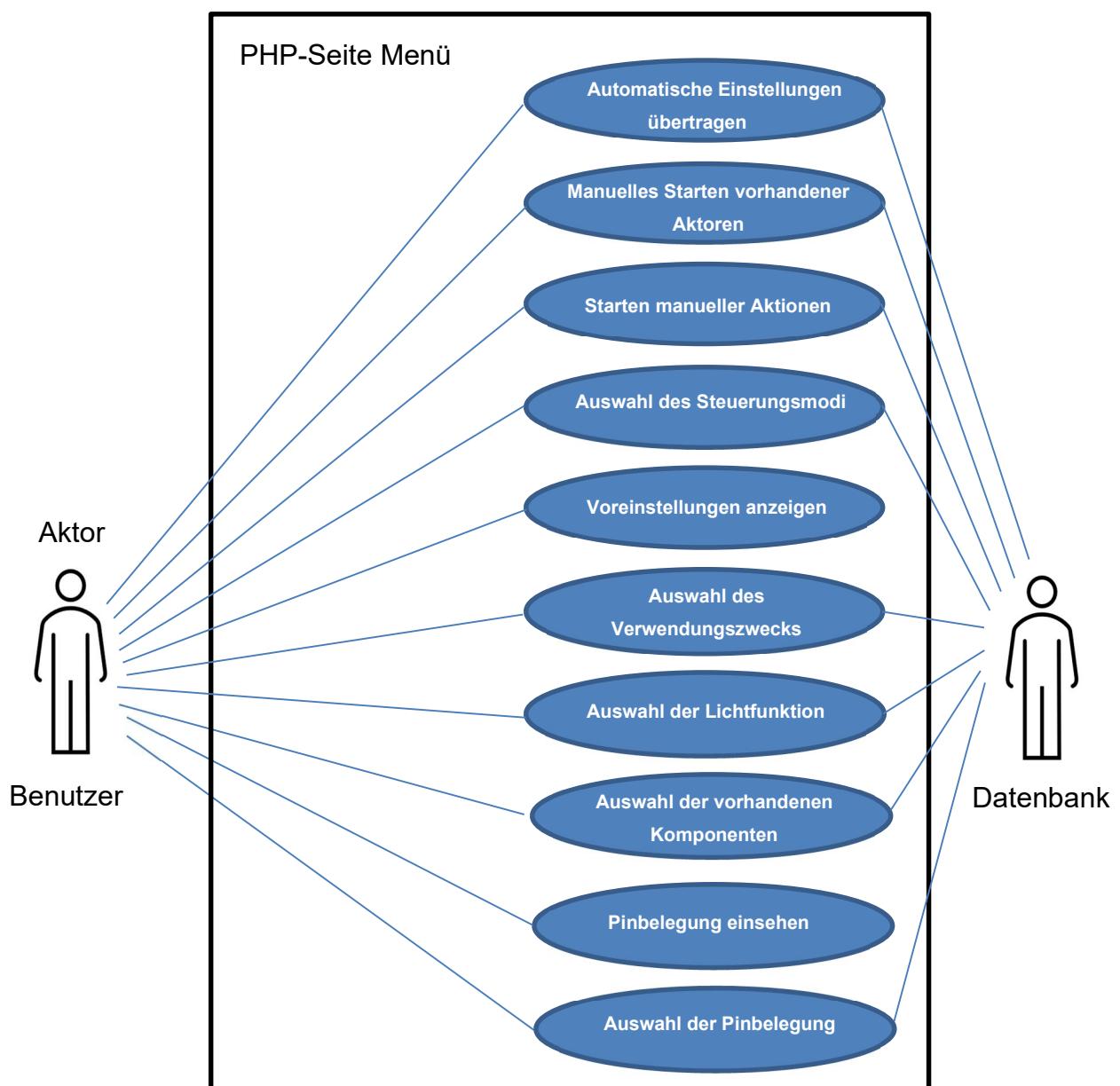


Abbildung 11: Use Case Diagramm Menüseite

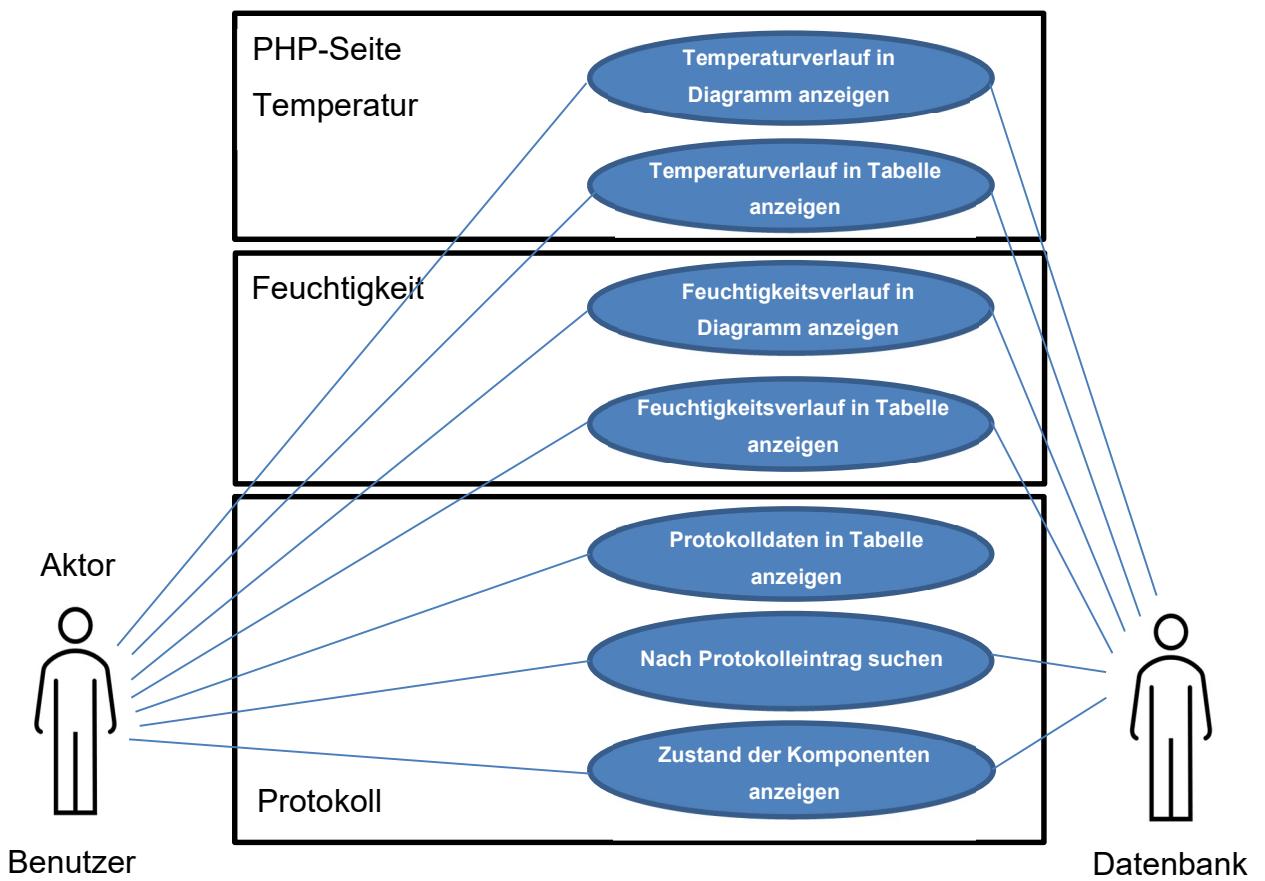


Abbildung 12: Use Case Diagramm Temperatur-, Feuchtigkeits -und Protokollseite

Die dokumentierten Use Cases befinden sich im Anhang 13.1.

2.1.3 Steuerprogramm

Das Steuerprogramm ist eine zentrale Applikation in diesem Projekt. Es dient dabei der Automatisierung der verschiedenen Prozesse.

Einstellparameter Bedingungen:

In Tabelle 1 befinden sich die Bedingungen für die Einstellparameter.

Mindestfeuchte	50 - 100%
Lichtstart	00:00 – 23:59 Uhr
Lichtende	00:00 – 23:59 Uhr (Nicht identisch mit Lichtstart)
Lüftungsintervall	Keine spezielle Vorgabe
Lüftungsdauer	5 - 30 min
Befeuchtungsdauer	5 - 30 min
Mindesttemperatur	0 - 25°C
Mindesttemperatur Nacht	0 - 25°C
Maximaltemperatur	15 - 35°C (Mindesttemperatur + 5°C)
Maximaltemperatur Nacht	15 - 35°C (Mindesttemperatur + 5°C)

Tabelle 1: Parameterbedingungen

Die Parameterbedingungen werden vom Java-Steuerungsprogramm im Zeitintervall von 3s von der Datenbank eingelesen. Beim Setzen der Parameter innerhalb des Java-Programms werden die Einstellparameter auf Plausibilität überprüft (Doppelte Absicherung: Anwender wird bereits auf der Steuerseite bei Falscheingabe mittels einer roten Markierung informiert). Bei einer fehlerhaften Prüfung werden die alten Werte beibehalten. Zusätzlich wird auch im Falle des manuellen Betriebs die Tabelle „manuell“ alle 3s ausgelesen. Dadurch ist es möglich die Aktoren im manuellen Betrieb mit einer maximalen Zeitverzögerung von 3s anzusteuern.

Startbedingungen:

Schalterstellung: Beim Start des Programms sind alle Manipulationskomponenten, wie Licht, Heizung, etc. standardmäßig ausgeschalten. Der DHT22 Temperatur- und Luftfeuchtigkeitssensor wird dagegen zu Beginn des Programms angeschaltet.

Default-Werte: In der untenstehenden Tabelle (Tabelle 2) befinden sich die Default-Werte, welche zu Beginn des Programms als Einstellparameter verwendet werden.

Mindestfeuchte	80%
Lichtstart	8:30 Uhr
Lichtende	21:30 Uhr
Lüftungsintervall	60 min
Lüftungsdauer	10 min
Befeuchtungsdauer	15 min
Mindesttemperatur	15°C
Maximaltemperatur	27°C

Tabelle 2: Default-Werte

Für die Maximaltemperatur und Minimaltemperatur bei Nacht werden innerhalb des Java-Programms keine extra Parameter angelegt. In diesem Fall wird beim Einlesen der Parameter aus der Datenbank darauf geachtet, ob momentan das Licht „an“ (Tag) beziehungsweise „aus“ ist (Nacht). Dies bedeutet, dass keine zusätzlichen Startparameter angelegt werden müssen. Wird das Licht auch als Heizung verwendet, so wird dies nicht als Bedingung für den Tag gewertet. In diesem Fall hat die Heizung Vorrang.

Messung:

Eine Messung findet im Normalbetrieb in Zeitabständen von 120 Sekunden statt. Nachdem die Daten über den Sensor eingelesen wurden, werden diese anschließend auf deren Richtigkeit überprüft. Falls die Daten Fehler aufweisen, so wird der Sensor neu gestartet. Folgen drei fehlerhafte Messungen in Folge, so wird das System in Notbetrieb versetzt. Im Notbetrieb findet lediglich alle 60 min eine erneute Messung statt (nachdem der Sensor zuvor neu gestartet wurde). Der Anwender wird in diesem Fall über die Protokollseite gebeten den DHT22 zu überprüfen.

Bei einer erfolgreichen Prüfung werden die Daten in die Datenbank geschrieben. Zusätzlich werden die Sensordaten mit den Einstellparametern (Mindesttemperatur, Höchsttemperatur, Mindestfeuchte) verglichen und bei Bedarf die jeweilige Manipulationskomponente (Heizung, Kühlung, Befeuchtungsroutine) eingeschalten. Genauerer Erläuterung der Ansteuerung befinden sich in den folgenden Abschnitten.

Sensorneustart:

Automatischer Betrieb:

Normalbetrieb: Im Normalbetrieb wird der Sensor, entsprechend den Projektanforderungen, alle 24h neugestartet. Dies findet immer um 24 Uhr statt. Dabei schaltet der entsprechende GPIO-Pin auf den Aus-Zustand (Sensor spannungsfrei). Anschließend läuft eine Wartezeit von 30 Sekunden ab, bis der Sensor wieder aktiviert wird.

Des Weiteren kann es auch aufgrund von fehlerhaften Messungen zu einem Neustart des Sensors kommen. Der Sensor wird dabei nach jeder fehlerhaften Messung neugestartet, bis das System nach drei fehlerhaften Messungen in Folge in den Notbetrieb versetzt wird.

Notbetrieb: Befindet sich das System im Notbetrieb, so wird der Sensor alle 60 min neu gestartet. Dabei wird der Sensor ausgeschalten und nach einer Wartezeit von 5 min erneut eingeschalten. Bei der nächsten erfolgreichen Messung wird letztendlich der Notbetrieb verlassen.

Manueller Betrieb:

Im manuellen Betrieb kann der Sensorneustart vom Anwender über die Betätigung eines Buttons auf der PHP-Steuerungsseite durchgeführt werden.

Hinweis: Nach jedem Sensorneustart (unabhängig vom Betrieb) wird nach Wiedereinschalten des Sensor 30s gewartet und anschließend eine Messung angestoßen. Befindet sich das System im Normalbetrieb so führt eine fehlerhafte Messung erneut zu einem Sensorneustart. Im Notbetrieb wird dagegen kein Sensorneustart eingeleitet.

Befeuchtung:

Automatischer Betrieb:

Normalbetrieb: Für den Start der Befeuchtungsroutine im Automatikbetrieb gibt es verschiedene Szenarien. Fällt die gemessene Luftfeuchtigkeit unter die vorgegebene Mindestfeuchtigkeit so wird im ersten Schritt die Lüftung über die Lüftungsdauer eingeschalten. Nach Ablauf der Lüftungsdauer wird eine Messung angestoßen. Befindet sich die gemessene Luftfeuchtigkeit erneut unter der gegebenen Mindestfeuchtigkeit so startet die Befeuchtungsroutine.

Ist dagegen ein Lüftungsintervall von 0s (Dauerlüften) vom Benutzer eingestellt und wird bei zwei aufeinanderfolgenden Messungen eine Luftfeuchtigkeit unter der Mindestfeuchtigkeit gemessen, so wird auch in diesem Fall eine Befeuchtungsroutine angestoßen. Bedeutet in diesem Fall hat die Befeuchtungsroutine eine höhere Priorität.

Notbetrieb: Wird das System in den Notbetrieb versetzt, so ist eine regelmäßige Befeuchtungsroutine mit einem Zeitintervall zwischen den Routinen von 15min eingestellt.

manueller Betrieb:

Im manuellen Betrieb können Fogger und Lüftung (niedertourig oder hochtourig) separat ein- und ausgeschalten werden. Zusätzlich kann eine Befeuchtungsroutine manuell angestoßen werden. Ist die Befeuchtungsroutine eingeschalten, so können Lüftung und Fogger nicht einzeln ausgeschalten werden.

Die Routine an sich beginnt mit dem Einschalten des Lüfters (niedertourig) und des Foggers. Nach Ablauf der Befeuchtungsdauer werden beide Komponenten wieder ausgeschalten.

Hinweis: Ist nur eine Betriebsart des Lüfters vorhanden, so wird diese Betriebsart bei der Befeuchtungsroutine verwendet.

Beleuchtung:

Automatischer Betrieb: Das Programm vergleicht im 3 Sekunden Takt die aktuelle Zeit mit den eingestellten Parametern Lichtstart und Lichtende. Im Zeitraum zwischen Lichtstart und Lichtende wird das Licht eingeschalten.

Ist beim Start des Programms ebenfalls die Option Licht als Heizung ausgewählt, so wird das Licht ebenfalls als Heizung verwendet (Verweis nächster Absatz).

Manueller Betrieb: Das Licht kann im manuellen Betrieb vom Nutzer über die Steuerungsseite (Toggle Switch) an und ausgeschalten werden.

Heizung:

Automatischer Betrieb: Die Heizung wird immer dann eingeschalten, wenn die momentane Temperatur unter die eingestellte Mindesttemperatur fällt. Die Temperatur wird dabei immer kontrolliert, wenn vom Sensor Daten ausgelesen werden (alle 120s).

Die Heizung wird ausgeschalten, wenn sich die Temperatur über der Schwelle „Mindesttemperatur + (Höchsttemperatur - Mindesttemperatur) *25%“ befindet.

Manueller Betrieb: Die Heizung kann vom Benutzer über die Steuerungsseite manuell ein- und ausgeschalten werden.

Kühlung:

Automatischer Betrieb: Die Kühlung wird immer dann eingeschalten, wenn die momentane Temperatur über die eingestellte Höchsttemperatur ansteigt. Die Temperatur wird dabei immer kontrolliert, wenn vom Sensor Daten ausgelesen werden (alle 120s).

Die Kühlung wird ausgeschalten, wenn sich die Temperatur unter der Schwelle „Höchsttemperatur - (Höchsttemperatur - Mindesttemperatur) *25%“ befindet.

Manueller Betrieb: Die Kühlung kann vom Benutzer über die Steuerungsseite manuell ein- und ausgeschalten werden.

Protokoll:

Protokolliert werden Fehlermeldungen, wie beispielsweise Wechsel in den Notbetrieb (falsche Messung des Sensors), unplausible Daten der Einstellungsparameter von der Steuerungsseite. Ebenfalls werden die Zustandsänderungen der Manipulationskomponenten, wie Licht, Heizung, etc., sowie den Wechsel von Manuell in Automatikbetrieb im Protokoll gespeichert. Dieses Protokoll ist auf der Steuerungsseite unter dem Reiter „Protokoll“ in Form einer Tabelle zu sehen. Zusätzlich liegt auf dem Raspberry Pi ein Ordner mit allen „log“-Dateien im Projekt-Ordner ab. Dabei wird für jeden Tag eine neue „log“-Datei erstellt, in welchem die Protokolldaten für den jeweiligen Tag gespeichert werden. Um Mitternacht wird die „log“-Datei in einen „History“-Ordner verschoben und dort als .zip-Datei abgelegt. (Konfiguration des Loggers log4j)

Wechsel vom manuellen in den automatischen Betrieb:

Beim Wechseln vom manuellen in den automatischen Betrieb werden alle Komponenten außer dem Sensor ausgeschalten. Des Weiteren wird bei aktiver Befeuchtungsroutine auch diese abgebrochen, um die Standartbedingungen für den Automatikbetrieb zu gewährleisten.

Fehlerbehandlung:

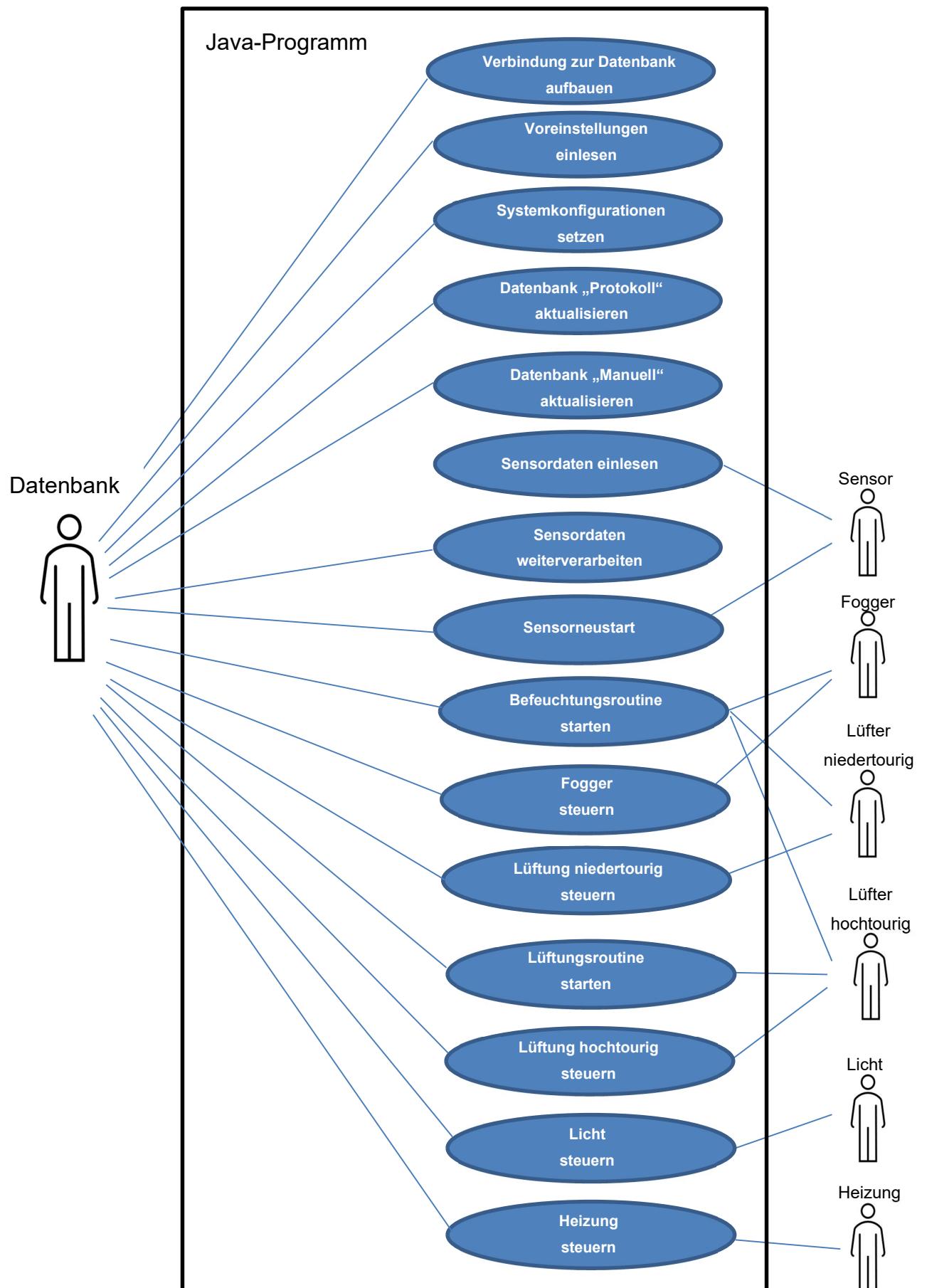
Alle Fehler, die während der Applikation auftreten können, werden sowohl in eine Datei, als auch in die Datenbanktabelle „protokoll“ aufgenommen und somit für den Anwender sichtbar. Je nach Schwere des Fehlers läuft das Programm weiter bzw. wird es abgebrochen. Ein Beispiel für ein Abbruch des Programms wäre eine fehlerhafte Verbindung zur Datenbank. Im Gegensatz dazu wird bei einer fehlerhaften Messung lediglich der Fehler protokolliert.

DHT22: In seltenen Fällen kann es zu Komplikationen beim Auslesen beziehungsweise beim Aufrufen des Python-Skripts des DHT22 kommen. Dabei werden keine Sensordaten eingelesen. In diesem Fall werden die alten Sensordaten beibehalten. Der Anwender wird dementsprechend über die Protokollseite informiert. Sollte es bei drei aufeinanderfolgenden Messungen zu solchen Komplikationen kommen, so wird der Anwender zusätzlich über die Protokollseite gebeten, den DHT22 zu prüfen. Das System wird hierbei nicht in Notbetrieb versetzt.

Zusatzinformationen:

Über die Datenbank „voreinstellungen“ werden zu Beginn des Programms einmal die vorhandenen Akteure sowie die Systemkonfiguration (Bsp. Licht als Heizung) des Systems eingelesen. Sollte eine Komponente nicht verfügbar sein, so ist die entsprechende Aktion ebenfalls nicht verfügbar. Beispielsweise wird keine Kühlung angestoßen, sofern kein Kühler im System vorhanden ist. Des Weiteren ist anzumerken das bei Änderung der Systemkonfiguration das Programm neugestartet werden muss.

2.1.4 Steuerprogramm Use Case Diagramm



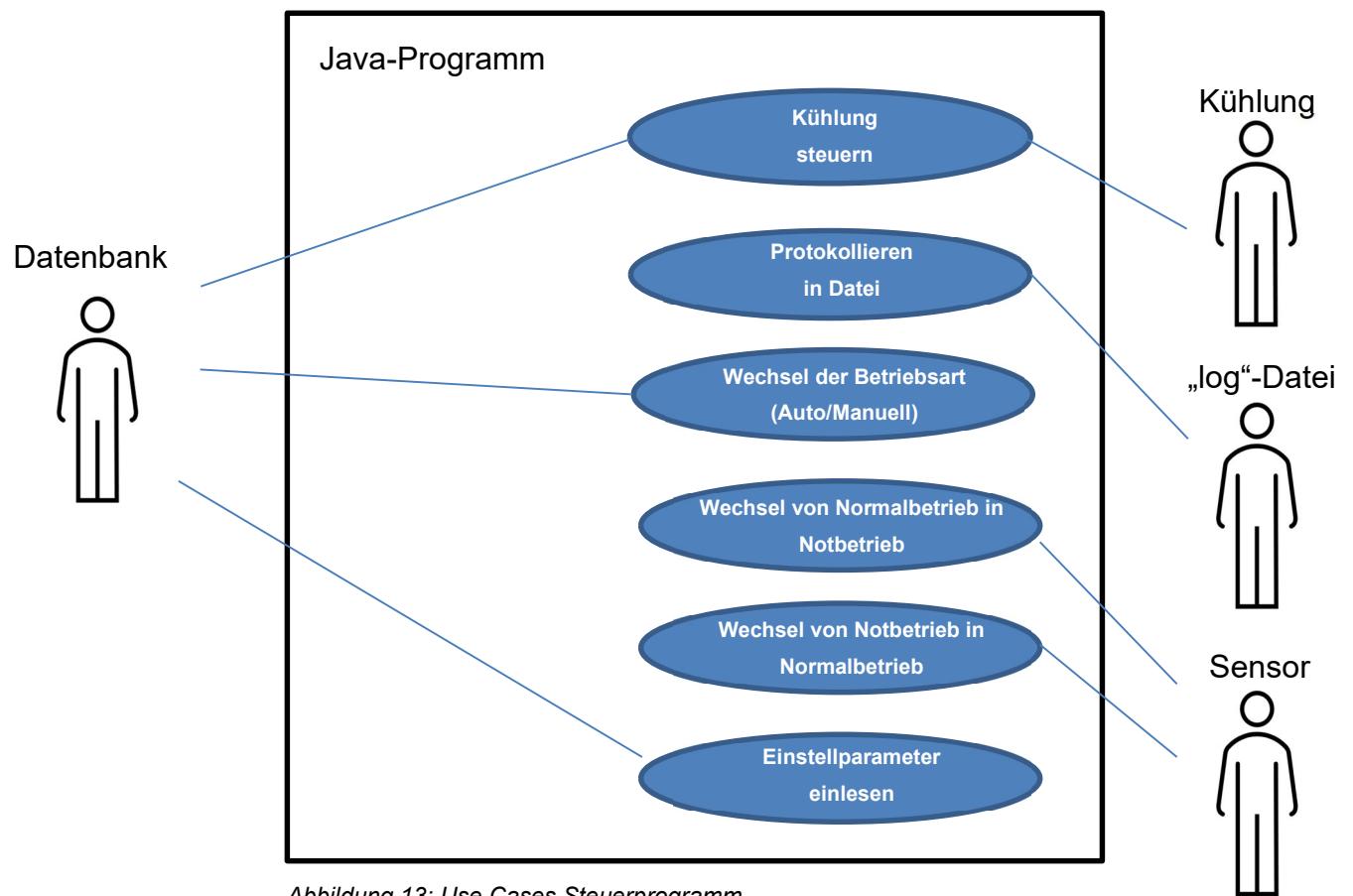


Abbildung 13: Use Cases Steuerprogramm

Die dokumentierten Use Cases befinden sich im Anhang 13.2.

2.2 Hardware

In der untenstehenden Tabelle 3 sind die verwendeten Komponenten zum Aufbau eines Hobby-Fungariums aufgelistet. Im Anhang 13.1 befinden sich die zugehörigen Internetlinks, unter denen die Komponenten käuflich zu erwerben sind:

Komponente	Im Projekt	Funktion
Lüfter	ebm W2S110-A001-39	Sorgt für eine aktive Belüftung und einen hinreichenden Sauerstoffgehalt
Fogger	Hygro-Plus Terrarien-Nebler	Sorgt für die notwendige Luftfeuchtigkeit
Sensor	DHT22	Messen der aktuellen Temperatur und Luftfeuchtigkeit
Relaisboard	Waveshare Raspberry Pi 8-ch Relays	Ansteuern der oben darüber aufgezählten Hardwarekomponenten
Raspberry Pi	Raspberry Pi 4 Modell B; 4 GB	Ansteuern des Relaisboards / Bereitstellen des Apache Webservers etc. → Funktionen im Laufe der Dokumentation beschrieben
Fungarium Gehäuse	Herkömmliches Aquarium aus Glas	Dient als Umweltumgebung für die Pilzzucht
Lichtquelle	Barthelme Glühlampe 230 V E27 15 W Glühlampenform	Versorgung der Pilze mit Licht
Steckdosen	PCE 601.450.06 Anbau-Steckdose IP54 Blau	Anschließen der Aktoren

Tabelle 3: Hardwarekomponentenliste

Zusätzlich zu den oben aufgelisteten Zukaufteilen werden weitere Komponenten, wie beispielsweise eine Verteilerbox zum Anschließen der Aktoren und Verstauen des Raspberry Pi und des Relaisboard, eigenständig gefertigt. In der oberen Liste befinden sich lediglich die Zukaufteile.

3 Port Weiterleitung

Vom internen „Heim-Netzwerk“ lässt sich der Apache Server mit der verwendeten IP-Adresse des Raspberry Pi im lokalen Netz (<http://192.168.X.XX>) erreichen.

Um den Apache-Server auch von einem externen Netzwerk aus erreichen zu können, bedarf es einer Port-Weiterleitung vom Heimrouter an das Raspberry Pi. Da die Konfiguration der Portweiterleitung für die darauffolgenden Einrichtungsschritte vorteilhaft ist, wird diese vorrangig behandelt.

Eine Portweiterleitung ist eine Netzwerktechnik, bei der eine Verbindung weitergeleitet wird. Sie wird auch Portzuordnung, Tunneling oder Punch Through bezeichnet. Dies geschieht über ein Rechnernetz auf einen bestimmten Port zu einem anderen Computer. Dabei wird die gesamte eingehende Kommunikation eines bestimmten Ports an denselben Port auf einem internen Netzwerknoten übertragen. Dadurch kann ein externes Quellennetzwerk eine Verbindung zu einem identischen Quellenknoten herstellen, der normalerweise eine Verbindung zu einem internen privaten LAN herstellt. Ein mit privat lokalem Netz und Internet verbundener Router, wartet an einem bestimmten Port auf Datenpakete. Wenn diese dort eintreffen, werden sie an einen anderen Port oder an einen Computer im lokalen Netz weitergegeben. Alle Datenpakete werden so verändert, dass es im Netz den Anschein hat, dass der Router die Pakete versendet (siehe Abbildung 14).¹

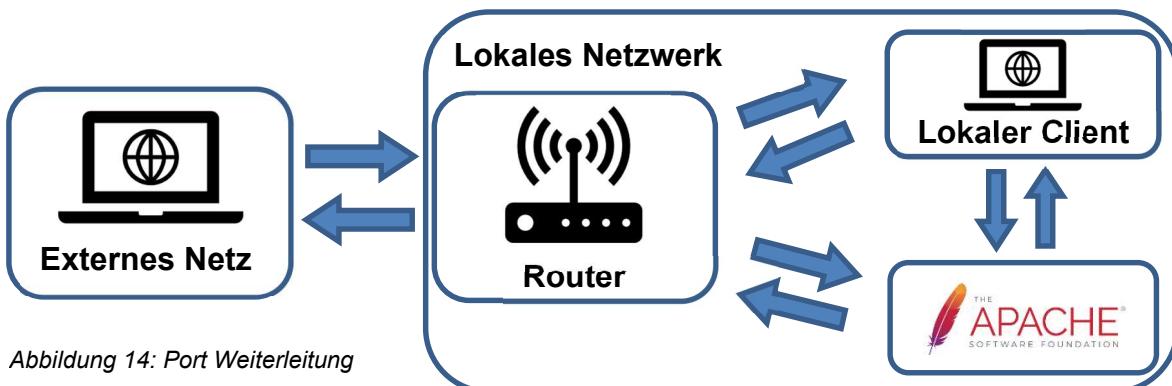


Abbildung 14: Port Weiterleitung

Allgemein gilt der Router als Schnittstelle zwischen dem im lokalen Netzwerk laufenden Apache-Webserver und dem externen Netz. Der lokale Webserver stellt im Projekt der Raspberry Pi dar, auf welchem der Apache Server läuft. Diese Weiterleitung ist wichtig, wenn der Server von einem externen Netzwerk aus aufgerufen werden soll. Um die Weiterleitung zu garantieren, müssen die jeweiligen Ports auf dem Raspberry Pi freigegeben werden.

¹ Vgl. (o.V., techopedia, o.J.)

3.1 Portfreigabe

Die Portfreigaben werden in den Router Einstellungen vorgenommen. Der Webserver in diesem Projekt wird unter einem „Fritzbox“- Router betrieben. Die Einstellung bei anderen Routern können abweichen.

Um die Geräteeinstellungen des Routers zu öffnen, wird vorerst in einem Webbrowesr die URL: „**fritz.box**“ eingegeben. Dabei ist es wichtig, dass sich das Gerät, mit welchem der Aufruf gemacht wird, im lokalen Netz des Routers befindet. Nachdem das korrekte Geräteword eingegeben wurde, wird sich die Startseite der Fritzbox-Einstellungen öffnen.

Unter dem Reiter „**Internet**“ befindet sich der Ordner „**Freigaben**“ (siehe Abbildung 15), in welchem die Portweiterleitung für das jeweilige Gerät, in unserem Fall der Raspberry Pi, eingestellt werden kann. Unter dem Reiter „**Portfreigaben**“ kann nun durch einen Buttonklick ein neues **Gerät für Freigaben hinzugefügt** werden.

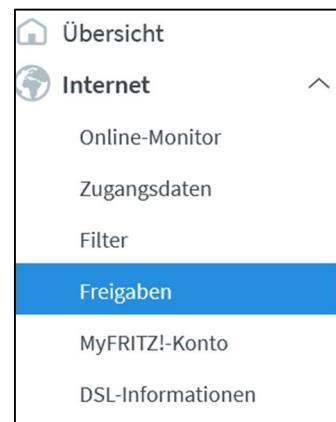


Abbildung 15: Freigaben Fritzbox

Als Gerät wird der Raspberry Pi, auf welchem der Apache Server läuft, ausgewählt (siehe Abbildung 16).

Wichtig ist, dass der Raspberry Pi



Abbildung 16: Geräteauswahl Portfreigabe

mit dem lokalen Netzwerk des Routers verbunden ist, da er sonst im Dropdown Menü „**Gerät**“ nicht zu finden ist.

Anschließend kann durch den Knopf am rechten unteren Bildrand „**Neue Freigabe**“ eine Portfreigabe für den Raspberry Pi erstellt werden.

Dieser Vorgang muss einmal mit der **Anwendung** als „**HTTP-Server**“ und „**HTTPS-Server**“ durchgeführt werden um die Ports „80“ und „443“ freizugeben. Nach der Bestätigung mit „**OK**“ sollte unter dem Reiter „**Portfreigaben**“ Folgendes zu sehen sein (siehe Abbildung 17).

raspberrypi	192.168.2.101 ::f3ba:e6c1:aee:82fc	HTTPS-Server HTTP-Server HTTPS-Server HTTP-Server	443 80 443 80
-------------	---------------------------------------	--	------------------------

Abbildung 17: Portfreigaben aktiviert

Damit ist die Konfiguration der Portfreigaben für den Raspberry Pi abgeschlossen.

4 Domain und DynDNS

Um im Webbrowser nicht ständig die IP-Adresse des Routers eingeben zu müssen, um die PHP-Steuerungsseite aufzurufen gibt es sogenannte Domains.

Jeder Webserver ist durch einen einzigartigen, numerischen Code gekennzeichnet. Diese sogenannte **IP-Adresse** ermöglicht es einem Browser, gezielt Hosts anzusprechen, um Webseiteninhalte abzurufen.

Bei einer Domain handelt es sich um einen **weltweit einmaligen, eindeutigen Namen** für einen logisch abgegrenzten Teilbereich des Internets – zum Beispiel eine Webseite. Internetnutzern begegnen Domains zum Beispiel in dieser Form:

„www.example.com“

Als wesentlicher Bestandteil eines „Uniform Resource Locators“ (URL) gibt die Domain an, wo eine Ressource innerhalb des hierarchisch strukturierten **Domain Name Systems (DNS)** zu finden ist. Die Übersetzung von Domains in IP-Adressen erfolgt durch sogenannte Nameserver. Dabei handelt es sich um spezialisierte Webserver, die mit der **Namensauflösung von IP-Adressen** betraut sind.¹

Für dieses Projekt ist es notwendig eine eigene Domain für den Webserver anzulegen, da auf diese Weise eine komfortablere Websuche des Servers möglich ist. Ebenso kann dadurch ein SSL-Zertifikat für die Domain ausgestellt werden, um eine sichere Verbindung zum Webserver aus dem externen Netz zu gewährleisten. Dabei gibt es verschiedene Arten die Rechte einer Domain zu erhalten und diese nutzen zu dürfen.

Dynamisches DNS oder **DDNS** ist eine Technik, um Domains im „Domain Name System“ (DNS) dynamisch zu aktualisieren. Das Problem ist, dass ein Router in unbestimmten Zeitabständen seine IP-Adresse wechselt. Der Zweck ist, dass durch DDNS nach dem Wechsel der IP-Adresse der Router automatisch und schnell den dazugehörigen Domaineintrag ändert. So ist der Router und daraus folgend der Raspberry Pi immer unter demselben Domainnamen erreichbar, auch wenn die aktuelle IP-Adresse für den Nutzer unbekannt ist. Dies ist sehr wichtig für den Zugriff aus einem externen Netzwerk auf den Raspberry Pi.²

¹ Vgl. (o.V., ionos, 2019)

² Vgl. (o.V., ionos, o.J.)

4.1 Domain und DynDNS über DDNSS

In diesem Projekt ist eine Domain verwendet, die über den Host „ddnss.de“ bereitgestellt und verwaltet wird.

Dabei muss zuerst ein kostenloser Account auf der Webseite „www.ddnss.de“ erstellt werden. Anschließend kann im User Dashboard nach dem Login des Nutzers unter dem Reiter „Domain Verwaltung“ die Seite „Neu Erstellen“ ausgewählt werden, um eine neue Domain zu erstellen (siehe Abbildung 18).

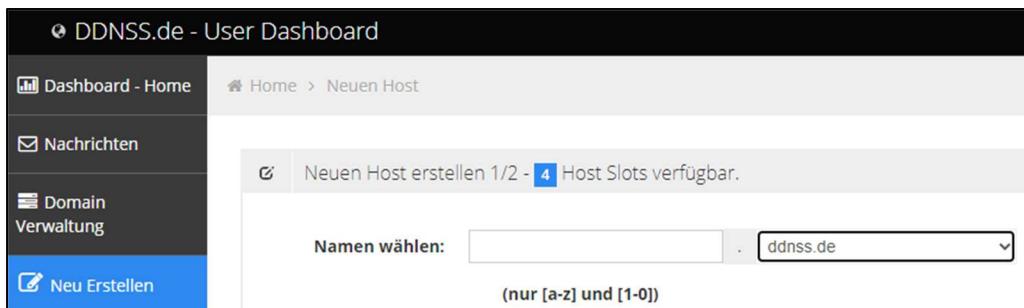


Abbildung 18: DDNSS Domain erstellen

Dabei ist es sinnvoll die Endung „.de“ auszuwählen, wenn die Seite in Deutschland betrieben wird. Der Name kann frei gewählt werden und wird mit „Weiter“ bestätigt. Im Folgenden wird, wie in Abbildung 19 gezeigt, der IP-Mode „A“ ausgewählt.

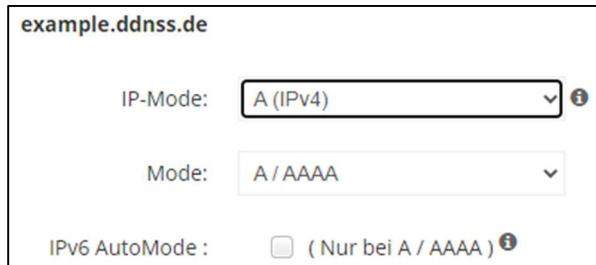


Abbildung 19: Konfiguration Domain

Die Domain ist daraufhin nach der Erstellung einsatzbereit.

Anschließend muss der DynDNS-Anbieter mit dem verwendeten Router verbunden werden, um die Domain auf den Router umzuleiten. Dazu muss unter „fritz.box“ unter dem Reiter „Internet“/ „Freigaben“ die Seite „DynDNS“ aufgerufen werden. Auf dieser Seite müssen folgende Anmeldedaten eingegeben werden (Abbildung 20):

DynDNS-Anbieter:	Benutzerdefiniert
Update-	
URL:	<a href="https://www.ddnss.de/upd.php?user=<username>&pwd=<pass>&host=<domain>">https://www.ddnss.de/upd.php?user=<username>&pwd=<pass>&host=<domain>
Domainname:	example.ddnss.de
Benutzername:	hier trägst Du dein Account Benutzername ein.
Kennwort:	hier trägst Du dein Account Kennwort ein.

Abbildung 20: Anmeldedaten DynDNS-Anbieter

5 Installation über Bash-Skript

Um die Installation und Konfiguration des Projekts (Installation und Einrichtung des Apache-Servers, der MySQL-Datenbank und des Java-Steuerungsprogramms) zu erleichtern, liegt dem Projekt ein Bash-Skript bei. Mit Hilfe des Bash-Skripts kann eine manuelle Eingabe der erforderlichen Kommandos vermieden werden. Das Bash besteht aus zwei Teilen. Diese Struktur ist gewählt, da während der Ausführung des Skriptes ein Neustart des Systems erforderlich ist. Durch den zweiteiligen Aufbau wird sichergestellt, dass die installierten Komponenten fehlerfrei auf dem Raspberry Pi laufen. Dabei ist es wichtig das Bash-Skript nach dem Neustart mit demselben Kommando (wie beim ersten Start) zu starten, sodass die restlichen Pakete heruntergeladen und installiert werden können!

Während der Installation der einzelnen Komponenten werden einige wichtige Eingaben abgefragt die jeweils vom Anwender eingegeben und mit der **ENTER**-Taste bestätigt werden müssen. Bei den Eingaben handelt es sich beispielsweise um die verwendete Domainadresse, die vor der Ausführung des Bash-Skripts vorhanden und eingerichtet sein muss. Dabei ist wichtig, dass die Portweiterleitung und das DynDNS funktionsfähig sind. Des Weiteren müssen E-Mail-Adresse, Benutzername und Passwörter der zu installierenden Komponenten eingegeben werden.

Hinweis: Bei der Eingabe der Passwörter ist darauf zu achten, dass die eingegebenen Zeichen aus Sicherheitsgründen nicht auf dem Bildschirm angezeigt werden.

Bevor das Bash-Skript ausgeführt werden kann, muss das Github Repository zusammen mit dem Java Steuerungsprogramm heruntergeladen werden. Dies erfolgt mit folgendem Kommando im Terminal:

```
sudo git clone https://github.com/svenSchelling/Fungarium.git  
/home/pi/Fungarium
```

Dabei wird der Ordner auf dem Hauptverzeichnis des Raspberry Pi: /home/pi abgelegt.

Um das Skript auszuführen, muss vorerst in den erstellten Ordner navigiert werden, in dem sich das Bash-Skript befindet:

```
cd /home/pi/Fungarium
```

Meist ist der Start nicht direkt möglich aufgrund der Berechtigung des jeweiligen Nutzers. Die Berechtigung kann über folgenden Command im Terminal verändert werden:

```
chmod 774 start_fungarium.sh
```

Der Startaufruf des Skripts wird anschließend über folgendes Kommando im Terminal auf dem Raspberry Pi ausgeführt:

```
./install_fungarium
```

Nach dem Neustart innerhalb des Bash-Skripts muss zur Ausführung des zweiten Skript-Teils, dasselbe Kommando eingegeben werden.

Das Bash-Skript ist speziell für dieses Projekt erstellt, kann aber mit geringem Aufwand für die einfache und schnelle Einrichtung eines Apache-Servers mit verbundener MySQL-Datenbank verwendet werden.

Wird die Installation anhand des Bash-Skripts durchgeführt muss die manuelle Installation der Komponenten, die von Kapitel 6 bis 10 anhand von einzelnen Kommandos aufgeführt ist, nicht ausgeführt werden. Diese Installationsschritte werden in dem automatisierten Bash-Skript abgearbeitet.

6 LAMP

Um einen Apache-Webserver einzurichten, wird die Installation von LAMP benötigt. Die Bezeichnung LAMP setzt sich aus den Anfangsbuchstaben des Betriebssystems Linux, des Webservers Apache, des Datenbanksystems MySQL und der Skriptsprache PHP zusammen. Dabei handelt es sich um eine Softwareumgebung für Serversysteme, mit denen sich statische und dynamische Webinhalte zur Verfügung stellen lassen. Dabei stellen die einzelnen Softwarekomponenten alle Funktionen bereit, die dafür benötigt werden, um Webanfragen anzunehmen, zu bearbeiten und Ergebnisse auszuliefern.¹

Diese einzelnen Komponenten verbunden mit dem Interface „PHPMyAdmin“ zur Verwaltung der MYSQL- Datenbank werden im folgenden Kapitel näher erläutert.

Dabei wird die Installation der Komponenten, ihre Besonderheiten und die jeweilige Verwendung im Projekt thematisiert.

6.1 Linux und Raspbian

Linux ist ein Open-Source-Betriebssystem, welches auf dem Linux-Kernel (Betriebssystemkern) und einer freien GNU-Software basiert.

Für den Betrieb des Raspberry Pi wird ein Betriebssystem benötigt. In der Regel wird man eine Linux-Distribution einsetzen. Raspbian ist der Klassiker unter den Linux-Distributionen für den Raspberry Pi. Raspbian basiert auf Debian, bei dem auf einen stabilen und sicheren Betrieb sehr viel Wert gelegt wird.

In diesem Projekt wird das Betriebssystem Raspbian verwendet, welches vor Beginn des Projekts auf der Micro-SD Karte des Raspberry Pi's installiert werden muss. Ein Betriebssystem ist die Voraussetzung, um die folgenden Installationen des Projekts ausführen zu können.

¹ Vgl. (Luber, 2019)

6.2 Apache Server

Apache ist ein kostenloser Open Source Webserver. Dieser Webserver zählt zu den populärsten der Welt. In diesem Projekt wird er dazu verwendet die Steuerungsseite des Fungariums bereitzustellen. Dabei soll in diesem Kapitel die Installation und Konfiguration des Apache- Servers auf dem Raspberry Pi im Zusammenhang mit der „.htaccess“- Datei näher erläutert werden.

6.2.1 Allgemeines

Apache wird aktuell von rund 46% aller Webseiten weltweit genutzt. Die erste Apache Version wurde im Jahre 1995 veröffentlicht. Verwaltet und entwickelt wird der Webserver von der Apache Software Foundation (ASF). „Apache http Server“ stellt dabei den vollständigen Namen des Webservers dar.¹

Bei einem Apache- Server handelt es sich um einen physischen Server. Dabei beschreibt Apache die plattformunabhängige Software, die sowohl auf Linux als auch auf Windows Servern ausgeführt werden kann. Die Aufgabe besteht darin eine Verbindung zwischen einem physischen Server mit den gespeicherten Webseiten und den Browsern des Internetnutzers herzustellen.

Der Ablauf eines Serveraufrufs wird anhand von Abbildung 21 dargestellt.

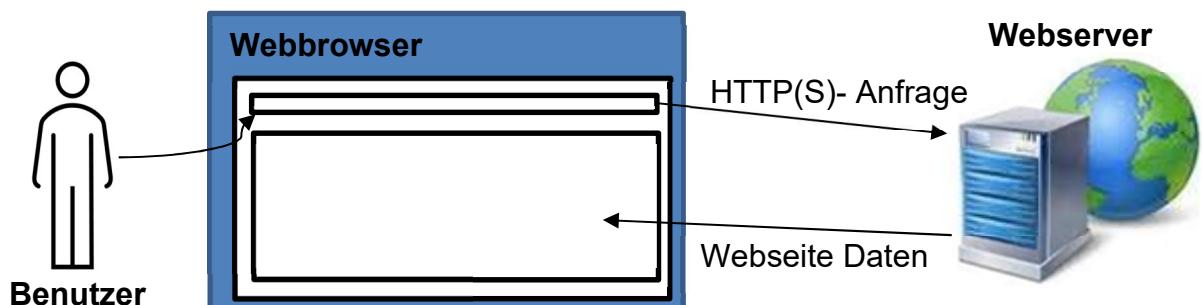


Abbildung 21: Schaubild Serveraufruf

Ein „User“ gibt eine „URL“ in seinen Webbrowser ein. Dabei sendet der Browser eine HTTP oder HTTPS Anforderung an den jeweiligen Server (in unserem Fall ein Apache Server), auf dem die Webseite gespeichert ist. Der Apache- Server, der auf dem Server installiert ist, verarbeitet die Anfrage und gibt die angeforderten Webseiten zurück.

¹ (o.V., biteno.com, o. J.)

6.2.2 Installation

In diesem Kapitel ist die Installation von „Apache2“ auf dem Betriebssystem Raspbian beschrieben.

Um diese Installation auszuführen wird ein Terminal benötigt, in welchem das folgende Kommando ausgeführt werden muss, um sicher zu stellen, dass alle Pakete sich auf dem neuesten Stand befinden.

```
sudo apt-get update
```

Anschließend wird der Apache2 heruntergeladen und installiert:

```
sudo apt-get install apache2 -y
```

Dabei wird der Apache- Server über den Paketmanager „apt“ installiert. Mit „sudo“ wird festgelegt, dass das Kommando vom Superuser „root“ ausgeführt werden soll.

Apache legt nach der Installation eine HTML- Datei im Webordner unter dem Pfad /var/www/html/ mit der Bezeichnung index.html ab. Um zu prüfen, ob die Installation erfolgreich verlaufen ist, muss diese Datei unter <http://localhost/> oder der IP- Adresse des Raspberry Pi (<http://192.168.X.XXX>) im jeweiligen Netzwerk zu sehen sein.

Die Datei index.html ist in Abbildung 22 dargestellt.¹

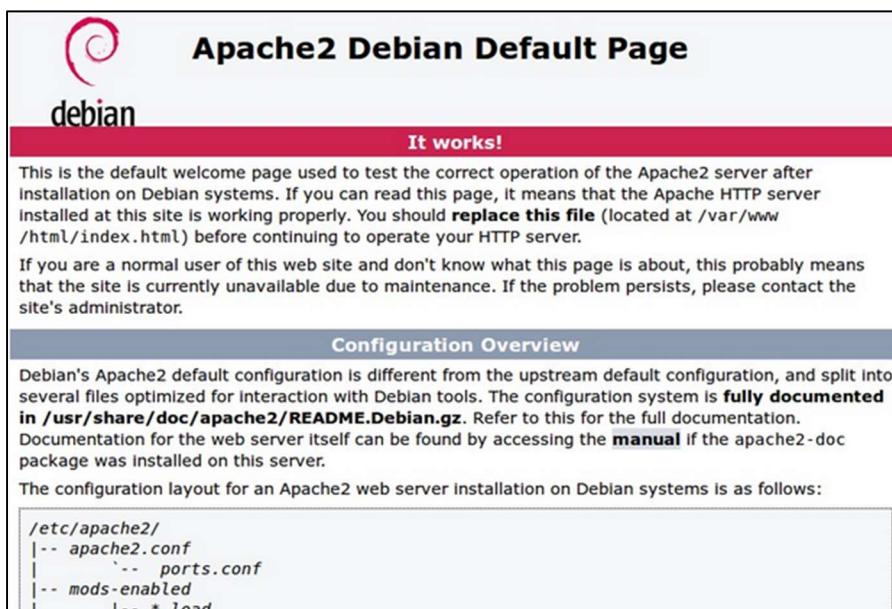


Abbildung 22: index.html Datei

¹ Vgl. (o.V., raspberrypi.org, o.J.)

6.3 Virtual Host

Für die eingerichtete Domain in den vorherigen Kapiteln wird nun ein Virtual Host erstellt, um diese mit dem Apache-Server zu verknüpfen.

Mithilfe eines Apache Servers kann ein Administrator einen Server einrichten, um mehrere Domains oder Standorte außerhalb einer einzigen Schnittstelle oder IP zu hosten, indem er ein übereinstimmendes System verwendet.

Jede Domain oder einzelne Website – bekannt als "virtueller Host" – die mit Apache konfiguriert ist, leitet den Besucher zu einem bestimmten Verzeichnis mit den Informationen der jeweiligen Seite. Die Informationen der Seite sind in diesem Projekt die PHP-Dateien, die auf der Steuerungsseite angezeigt werden. Die Basiseinheit, die einen einzelnen Standort oder eine Domain beschreibt, wird als „virtueller Host“ bezeichnet.

6.4 Erstellung Virtual Host

Bevor der Virtual Host erstellt werden kann, muss die Domainadresse in die Hostdatei des Raspberry Pi eingebunden werden, sodass die Domain innerhalb sowie außerhalb des Heimnetzwerks erreichbar ist. Dazu muss folgender Befehl im Terminal ausgeführt werden:

```
sudo echo "0.0.0.0 example.ddnss.de" | sudo tee -a /etc/hosts
```

Im **ersten Schritt** ist eine Verzeichnisstruktur zu erstellen, die die Website-Daten enthält, die wir den Besuchern zur Verfügung stellen. Das Verzeichnis der obersten Ebene, das Apache betrachtet, um Inhalte zu finden wird auf ein Verzeichnis unter `/var/www` festgelegt. In diesem Verzeichnis wird ein `public_html` Ordner erstellt, der die PHP-Dateien enthält. Dadurch wird eine gewisse Flexibilität im Hosting erreicht.

Das Kommando, um dieses Verzeichnis zu erstellen wird im Folgenden abgebildet. Dabei ist bevorzugt der Verzeichnisname der jeweiligen Domain auszuwählen.

```
sudo mkdir -p /var/www/example.ddnss.de/public_html
```

Im **Schritt zwei** sollten die Berechtigungen geändert werden, um sicherzustellen, dass der Lesezugriff auf das allgemeine Webverzeichnis und alle darin enthaltenen Dateien und Ordner zulässig sind, damit Seiten korrekt bedient werden können:

```
sudo chmod -R 777 /var/www
```

Im **dritten Schritt** sollen die Daten für die Steuerungsseite von Github in den Ordner „public_html“ geladen werden. Dazu muss zuerst sichergestellt, dass „git“ auf dem Raspberry Pi installiert ist. Die Installation erfolgt über den folgenden Befehl:

```
sudo apt install git
```

Um die Daten in den richtigen Ordner zu laden, muss vorerst in den Ordner „public_html“ navigiert werden:

```
cd /var/www/example.ddnss.de/public_html
```

Anschließend kann mit „git clone“ das Repository von Github geladen werden:

```
sudo git clone  
https://github.com/svenSchelling/FungariumPHPSteuerungsseite.git
```

Der Code für die PHP-Steuerungsseite ist unter folgender URL erreichbar:

<https://github.com/svenSchelling/FungariumPHPSteuerungsseite>

Im vierten Schritt wird die „Virtual Host“- Datei erstellt. Virtuelle Hostdateien sind die Dateien, die die tatsächliche Konfiguration des virtuellen Hosts angeben und bestimmen, wie der Apache-Webserver auf verschiedene Domainsanfragen reagieren wird.

Apache kommt mit einer standardmäßigen virtuellen Hostdatei namens „000-default.conf“, die zum Teil als Ausgangspunkt verwenden werden kann.

Zum Erstellen der Virtuellen Hostdatei muss deswegen die „default“-Datei kopiert werden:

```
sudo cp /etc/apache2/sites-available/000-default.conf  
/etc/apache2/sites-available/example.ddnss.de.conf
```

Dabei ist sehr wichtig, dass das“.conf“ am Ende des Domainnamens nicht vergessen wird.

Anschließend kann die Datei mit dem Texteditor „nano“ geöffnet werden:

```
sudo nano /etc/apache2/sites-available/example.ddnss.de.conf
```

Die Datei sollte nach dem folgenden Schema angepasst werden, um mit den folgenden Konfigurationsschritten weiterzuarbeiten. Dabei sollte jeweils der Namen der Domain und die E-Mail-Adresse des Server Admin angepasst werden.

```
<VirtualHost example.ddnss.de:80>
    ServerAdmin admin@example.com
    ServerName example.ddnss.de
    ServerAlias www.example.ddnss.de
    DocumentRoot /var/www/example.ddnss.de/public_html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    RewriteEngine on
    RewriteCond %{SERVER_NAME} =example.ddnss.de
    RewriteRule ^ https:// %{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>
```

Daraufhin muss mit der Tastenkombination **STRG + O** und anschließend **ENTER** die Datei gespeichert werden. Danach kann der Texteditor „nano“ mit **STRG + X** geschlossen werden.

Im fünften Schritt muss die erstellte Hostdatei aktiviert werden. Dies kann über folgendes Kommando durchgeführt werden:

```
sudo a2ensite example.ddnss.de.conf
```

Daraufhin muss Apache neu gestartet werden, damit die Änderungen der Hostdatei wirksam werden:

```
sudo systemctl restart apache2
```

Bei einer erfolgreichen Konfiguration müsste nun die Steuerungsseite unter „<http://example.ddnss.de>“ zu sichtbar sein (siehe Abbildung 23).¹

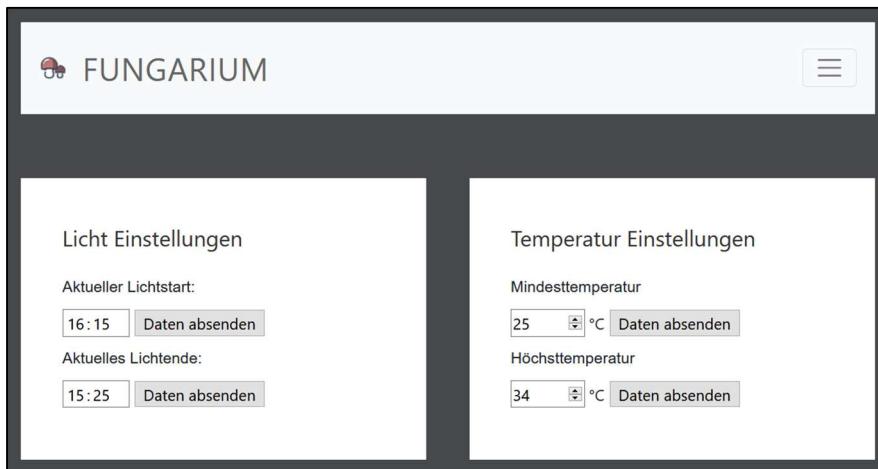


Abbildung 23: Steuerungsseite Menü

¹ Vgl. (o.V., pimylifeup.com, 2019)

6.5 .htaccess und .htpasswd

Für den Apache- Server stehen zahlreiche Module zur Verfügung, mit denen der Funktionsumfang des Servers erweitert werden kann. Darunter fällt beispielsweise eine Passwortauthentifizierung unter Verwendung einer „.htaccess“- Datei.

Eine „.htaccess“- Datei ist eine Server- Konfigurationsdatei für Verzeichnisse. Diese Technik ist der übliche Weg für bestimmte Benutzer des Webservers Zugriff auf bestimmte Daten zu erlauben

6.5.1 Einbinden der „.htaccess“- Datei

Die Voraussetzung einer „.htaccess“- Datei einzubinden ist, dass ein Apache- Server bereits lauffähig auf dem jeweiligen System vorliegt. Bevor die „.htaccess“- Datei erstellt werden kann, muss der Apache- Server so konfiguriert werden, dass er bei einem Serveraufruf nach einer „.htaccess“- Datei sucht und diese abarbeiten kann. Dies macht der Server standartmäßig nicht. Deswegen muss die Konfigurationsdatei des Apache- Servers dementsprechend bearbeitet werden. Diese Datei mit der Bezeichnung `apache2.conf` ist nach der Apache Installation in dem Ordner `/etc/apache2` zu finden. Um die Datei mit dem Texteditor „nano“ zu öffnen, muss folgendes Kommando im Terminal eingegeben werden:

```
sudo nano /etc/apache2/apache2.conf
```

Ist die Datei geöffnet muss die jeweilige „Directory“ ausgewählt werden. Die „Directory“ beschreibt den Ordnerpfad, in welchem die „.htaccess“- Datei folglich gespeichert wird. In unserem Fall handelt es sich um die „Directory“ `/var/www/example.ddnss.de/`.

```
<Directory /var/www/example.ddnss.de/>
    Options Indexes
    FollowSymLinks
    AllowOverride AuthConfig
    Require all granted
</Directory>
```

In diesem Element muss das Argument von „AllowOverride“ von „None“ zu „AuthConfig“ geändert werden. Dies ist das Kommando für den Apache Server, dass er bei einem Aufruf nach einer „.htaccess“- Datei sucht. Bei Bedarf kann die Bezeichnung „.htaccess“ in „AccessFileName“ geändert werden. Dies ist für den weiteren Verlauf der Installation nicht von Vorteil.

Daraufhin muss mit der Tastenkombination **STRG + O** und anschließend **ENTER** die Datei gespeichert werden. Danach kann der Texteditor „nano“ mit **STRG + X** geschlossen werden.

Um die Konfiguration des Apache Servers zu aktualisieren, muss der Server neu gestartet werden. Dies erfolgt über das folgende Kommando im Terminal:

```
sudo systemctl restart apache2
```

Nach der Konfiguration des Apache Servers kann die „.htaccess“- Datei erstellt werden. Dies erfolgt über das folgende Kommando:

```
sudo nano /var/www/example.ddnss.de/ .htaccess
```

Dabei ist es wichtig die Datei in dem Ordner zu erstellen, der bei der vorherigen Konfiguration angegeben ist (grün hinterlegter Text).

Mit dem Kommando wird die „.htaccess“- Datei erstellt und mit dem Editor „nano“ geöffnet. Nun muss in die leere Datei die folgenden Zeilen eingetragen werden:

```
AuthType Basic  
AuthName "Sicherer Bereich!"  
AuthUserFile  
/var/www/example.ddnss.de/.htpasswd  
Require valid-user
```

In der ersten Zeile wird festgelegt, dass ein Authentifizierungstyp vom Wert „Basic“ benutzt wird. Dies bedeutet, dass über eine Datei, in unserem Fall die Datei „.htpasswd“ authentifiziert wird. Mit „AuthName“ kann eine Meldung gewählt werden, die beim Serveraufruf angezeigt wird. Diese kann bei Belieben geändert werden. Mit „AuthUserFile“ wird die Authentifizierungsdatei und damit verbunden ihr Ordnerpfad angegeben. Um eine höhere Sicherheit zu gewährleisten, kann die Datei bei Belieben in einem anderen Ordner gespeichert werden. Dies ist nicht zwingend notwendig, da die Datei „.htpasswd“, die gespeicherten Passwörter verschlüsselt.

Im Folgenden muss mit der Tastenkombination **STRG + O** und anschließend **ENTER** die Datei gespeichert werden. Danach kann der Texteditor „nano“ mit **STRG + X** geschlossen werden.

Im nächsten Schritt erstellen wir die Sicherheitsdatei „.htpasswd“, in welcher folglich die Nutzer- Passwort Paare gespeichert werden. Dies geschieht über folgendes Kommando:

```
sudo htpasswd -c /var/www/example.ddnss.de/.htpasswd Nutzer
```

Mit diesem Kommando wird die Datei „.htpasswd“ erstellt und gleichzeitig ein erster Nutzer angelegt. Die Bezeichnung des Nutzers ist in diesem Falle beliebig auszuwählen. Nach der Ausführung des Kommandos wird nach einem Passwort des erstellten Nutzers gefragt, welches aus Sicherheitsgründen wiederholt eingegeben werden muss.

Anschließend kann mit diesem Nutzer- Passwort Paar auf den Webserver zugegriffen werden und die Installation eines passwortgeschützten Webservers ist abgeschlossen.

Soll nun nachträglich ein weiterer Nutzer für den Webserver erstellt werden muss im Terminal folgendes Kommando ausgeführt werden:

```
sudo htpasswd /var/www/example.ddnss.de/.htpasswd Nutzer2
```

Um nachträglich zu überprüfen welche Nutzer in der Datei „.htpasswd“ verifiziert sind, kann mit dem folgenden Kommando die Datei geöffnet werden:¹

```
sudo nano /var/www/example.ddnss.de/.htpasswd
```

¹ Vgl. (Ellingswood, 2015)

6.6 HTTPS und SSL

Der Begriff **SSL** (kurz für „Secure Socket Layers“) bezeichnet eine Technik, die man zur **Verschlüsselung und Authentifizierung des Datenverkehrs** im Netz einsetzt. Damit wird bei Webseiten die Übertragung zwischen Browser und Webserver gesichert. Immer wenn vertrauliche und sensible Daten übertragen werden, ist der Einsatz eines SSL-Zertifikats bzw. der Weiterentwicklung TLS („Transport Layer Security“) unumgänglich.¹

Mit SSL/TLS stellt man sicher, dass die Kommunikation weder mitgelesen noch manipuliert werden kann und persönliche Daten nicht in die falschen Hände geraten. Mit der Abkürzung **HTTPS** („Hypertext Transport Protocol Secure“) ist das entsprechende **Protokoll zur sicheren Datenübertragung gemeint**. HTTP bezeichnet die nicht abgesicherte Variante, bei welcher theoretisch alle übertragenen Daten von Angreifern mitgelesen werden können. HTTPS überträgt die Daten hingegen verschlüsselt und stellt die Authentizität der Anfragen sicher. Das funktioniert über das SSL- bzw. TLS-Zertifikat. In Abbildung 24 soll der Aufruf einer SSL-verschlüsselten Webseite verdeutlicht werden.²



Bei dem Umstellen einer Webseite auf SSL/TLS ist man auf ein SSL/TLS Zertifikat von einer Zertifizierungsstelle angewiesen. Seit ihrer Einführung im Jahr 2015 steht hierbei mit den Zertifikaten der Non-Profit-Organisation **Let's Encrypt** eine kostenfreie, einfach zu installierende Alternative zu den klassischen kostenpflichtigen Zertifikaten zur Verfügung. Let's Encrypt bietet um ein SSL/TLS- Zertifikat für die gewünschte Domain zu beziehen das „Automatic Certificate Management Environment“ (ACME) **Cerbot** an. Zum einen können darüber die Inhaberschaft der Internet-Domain geprüft und Zertifikate automatisch ohne Ausfallzeit erstellen und installieren werden, um HTTPS bei den jeweiligen Webseiten freizuschalten.¹

¹ Vgl. (o.V., ionos.de, 2020)

² Vgl. (o.V., fuseon-media.com, 2018)

6.6.1 Installation ACME Certbot

Um ein SSL- Zertifikat für eine spezielle Domain zu erhalten, muss die HTTP- Webseite bereits auf dem Apache- Server implementiert sein und laufen. Ebenfalls muss der Standard HTTP- Port 80 in den Einstellungen des Routers zum jeweiligen Server, in unserem Fall der Raspberry Pi, weitergeleitet werden. Diese Voraussetzungen sollten erfüllt sein, soweit die Installationen, Konfigurationen etc. der vorherigen Kapitel keine Fehler aufweisen.

Um Certbot zu installieren und damit verbunden das SSL- Zertifikat für die gewünschte Domain zu erhalten, muss vorerst der Paket Manager „snapd“ installiert werden. Ein Snapd ist ein Bundle einer App und ihrer Abhängigkeiten, welches ohne Änderungen in vielen verschiedenen Linux-Distributionen funktioniert. Um dieses Paketformat zu installieren, muss folgendes Kommando im Terminal ausgeführt werden:

```
sudo apt install snapd
```

Nach der erfolgreichen Installation muss das System neugestartet werden, um Änderungen die Funktion von „snapd“ im System zu garantieren. Dies kann ebenfalls über das Terminal ausgeführt werden.

```
sudo reboot
```

Nach dem Neustart des Systems muss sichergestellt werden, dass die aktuelle Version von „snapd“ vorliegt. Dies erfolgt durch dieses Kommando:

```
sudo snap install core; sudo snap refresh core
```

Nun kann Certbot mit folgendem Kommando installiert werden:

```
sudo snap install --classic certbot
```

Nach der erfolgreichen Installation muss mit dem folgenden Kommando die Ausführung des „Certbot-Kommandos“ vorbereitet werden.

```
sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

Nun ist es möglich das SSL- Zertifikat für die gewünschte Domain mit folgendem Kommando zu aktivieren.

```
sudo certbot --apache
```

Der Vorteil der Certbot-Pakete ist, dass sie mit einem Cron-Job oder einem Systemd-Timer geliefert werden, der die SSL-Zertifikate automatisch erneuert, bevor sie ablaufen. Aufgrund dessen muss Certbot nicht erneut ausgeführt werden, es sei denn, die Konfiguration des Servers wird verändert.

Nach der erfolgreichen Aktivierung des SSL-Zertifikats für die gewünschte Domain, kann nun in der Router Konfiguration der Port 443 auf den Server, in diesem Falle der Raspberry Pi, umgeleitet werden, sodass der HTTPS-Server von außerhalb des Heimnetzwerks erreicht werden kann. Der Port 443 stellt dabei den Standard-Port für einen HTTPS-Server dar.

Wichtig ist ebenso, dass der Port 80 nicht geschlossen wird, da sonst ein automatisches Aktualisieren des SSL-Zertifikats nicht gewährleistet werden kann.¹

7 PHP

PHP ist eine Abkürzung für „PHP: Hypertext Preprocessor“, eine Open Source Skriptsprache speziell für Webentwicklungen. Es ist Code, der ausgeführt wird, wenn der Server eine Anfrage für eine Webseite über einen Webbrower erhält. Es wird ausgeführt, was auf der Seite angezeigt werden muss, und sendet diese Seite dann an den Browser. Im Gegensatz zu statischem HTML kann PHP unter verschiedenen Umständen unterschiedliche Inhalte anzeigen.

Das Ziel der Skriptsprache ist es dem Webentwickler die Möglichkeit zu geben schnell dynamisch generierte Webseiten zu erstellen.²

Die Sprache PHP wird in diesem Projekt dazu verwendet, um Datenbankanfragen an eine MySQL Datenbank zu machen, um die Steuerungsseite mit dem Java-Steuerungsprogramm zu verbinden. Da diese Aufgaben in reinem HTML nicht möglich sind, wird PHP zur Hilfe genommen.

7.1 Installation

Auf dem Raspberry Pi wird das PHP-Paket mit dem folgenden Kommando installiert:

```
sudo apt-get install php -y
```

¹ Vgl. (o.V., certbot.eff.org, o.J.)

² Vgl. (o.V., php.net, o.J.)

8 Datenbanksystem

Um die PHP-Konfigurationsseite mit dem Java-Steuerungsprogramm verbinden zu können wird ein Datenbanksystem als Schnittstelle benötigt.

Ein Datenbanksystem ist ein elektronisches Verwaltungssystem, das gesammelte Daten zu einer logischen Einheit verknüpft. Es ist sehr praktisch, um Datenbestände zu verwalten und die Abfrage bestimmter Informationen zu verwalten. Ein Datenbanksystem muss besonders mit großen Datenmengen effizient, widerspruchsfrei und dauerhaft umgehen können. Ein weiterer Vorteil ist, dass Rechte festgelegt werden können, um bestimmten Nutzern Zugriff auf spezielle Daten zu erlauben.

Wenn von einer Datenbank gesprochen wird, wird meist das jeweilige Datenbanksystem gemeint. Dies beinhaltet, wie in Abbildung 25 dargestellt, ein Datenbank-Managementsystem (DBMS) und eine oder mehrere Datenbanken (DB).

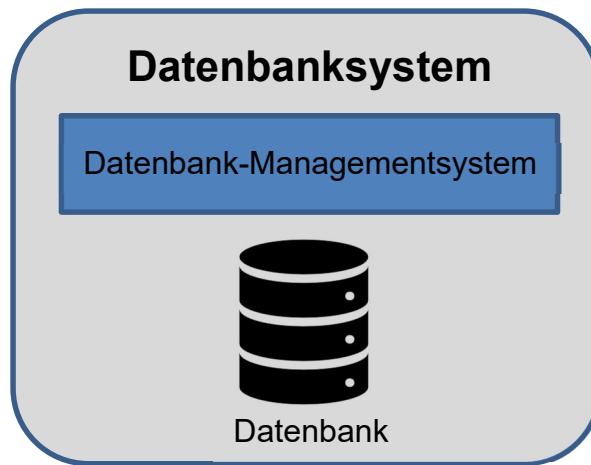


Abbildung 25: Schaubild Datenbanksystem

Dabei beschreibt die Datenbank die komplette Menge der zu ordnenden Daten. Diese Daten werden in Form von einzelnen Datensätzen abgelegt. Das Datenbank-Managementsystem ist für die Verwaltung der Daten verantwortlich und bestimmt die Struktur, die Ordnung, Zugriffsrechte, etc. Für die Verwaltung der Daten in einem DBMS liegt eine Datenbanksprache und ein geeignetes Datenbankmodell vor. In diesem Projekt wird mit dem DBMS MySQL/MariaDB das „Relationale Datenbankmodell“ verwendet. Für diese Art von DBMS wird das Akronym RDBMS benutzt, welches in unserem Beispiel in Verbindung mit der Datenbanksprache „SQL“ verwendet wird.¹

¹ Vgl. (o.V., datenbanken-verstehen.de, o.J.)

Das relationale Datenbankmodell arbeitet mit einzelnen Tabellen, in welchen Verknüpfungen der Daten festgelegt sind. Eine solche Beispieltabelle ist in Abbildung 26 dargestellt.

ID	Bezeichnung	Wert	letzte Änderung
1	Lichtstart	13:15:00	2020-11-01 17:57:37
2	Lichtende	14:16:00	2020-11-04 13:58:59

Abbildung 26: Beispieltabelle RDBMS

Eine Zeile dieser Tabelle bildet einen Datensatz, der auch als Tupel bezeichnet werden kann. Einzelne Informationen werden als Attribute (grüne Kasten) in den Spalten gesammelt. Die Gesamtrelation ergibt sich somit aus den zusammenhängenden Attributen (gelber Kasten). Dabei ist jedem Attribut ein spezieller Datentyp zugewiesen. Meist wird das erste Attribut einer Tabelle als Primärschlüssel festgelegt, sodass eine eindeutige Identifizierung eines Datensatzes gewährleistet werden kann. Dieser Primärschlüssel wird in diesem Beispiel mit der Bezeichnung „ID“ gekennzeichnet.

8.1 MySQL/MariaDB

MySQL ist ein quelloffenes Datenbank-Managementsystem, welches die Grundlage für dynamische Webseiten liefert. Im folgenden Kapitel wird MySQL zusammen mit dem im Projekt verwendeten DBMS MariaDB näher beschrieben. Folgend wird kurz auf die Verwaltungsoberfläche PHPMyAdmin, seine Installation und Verwendung im Projekt eingegangen.

8.1.1 Allgemeines zu MySQL/MariaDB

MySQL wurde 1994 vom schwedischen Unternehmen MySQL AB entwickelt. Das Datenbank- Managementsystems ist eine Open-Source- Software und universell auf nahezu jeder Plattform einsetzbar. Im Internet ist MySQL eines der meistverwendeten Datenbanksysteme. MySQL ist in der Lage große Datenmengen schnell zu verarbeiten und die Daten werden so gespeichert, sodass sie möglichst wenig Speicherplatz in Anspruch nehmen.

Der Name setzt sich aus „My“ und „SQL“ zusammen. „My“ steht dabei für den Namen der Tochter eines Mitgründers und „SQL“ (Structured Query Language) für die weltweit gebräuchlichste standardisierte Sprache für den Zugriff auf Datenbanken. Mit dieser Datenbanksprache kann die Datenbank angesteuert werden.¹

Nachdem der frühere Gründer das Softwareprojekt verließ, konzentrierte er sich fortan auf die Entwicklung des MySQL-Forks MariaDB, welches in diesem Projekt seine Anwendung findet.

8.1.2 Funktionsprinzip MySQL/MariaDB

MySQL arbeitet, wie in Abbildung 27 dargestellt nach dem Client-Server- Prinzip und besteht aus einem Server und einem oder mehreren Clients.

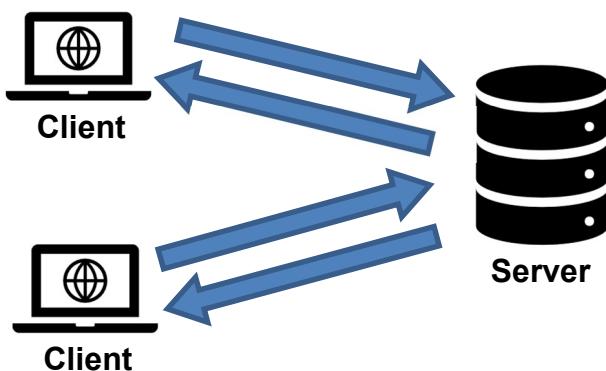


Abbildung 27: Schaubild Funktionsprinzip MySQL

Dabei stellt der Server das Datenbankmanagementsystem dar. Dieser sorgt für Verteilung und Speicherung von Daten. Innerhalb der Datenbank können zweidimensionale Tabellen mit unterschiedlichen Datentypen angelegt werden, in welchen sich die Datensätze befinden. Diese Daten können folglich mit Hilfe von der Datenbanksprache SQL bearbeitet werden.¹

8.1.3 Installation MariaDB

Die Pakete MariaDB Server und PHP-MySQL werden folgendermaßen installiert:

```
sudo apt-get install mariadb-server php-mysql -y
```

Anschließend muss der Apache-Server neu gestartet werden:

```
sudo service apache2 restart
```

¹ Vgl. (Luber, 2019)

8.2 PHPMyAdmin

Für Datenbanken werden häufig Verwaltungsoberflächen entwickelt, um dem Benutzer einen einfacheren Überblick und eine erleichterte Arbeitsweise zu verschaffen. Die bekannteste Oberfläche für MySQL/MariaDB ist „phpMyAdmin“.



Abbildung 28: Schaubild phpMyAdmin

Für dieses Projekt ist die Installation von PHPMyAdmin nicht zwingend notwendig, da SQL-Aufrufe zur Datenbank auch über das Terminal gemacht werden können. Durch die Installation wird aber für den Entwickler eine schöne Alternative geboten, um seine Datenbanken bequem über eine Oberfläche verwalten zu können. Ist die Installation von PHPMyAdmin nicht gewünscht, muss nur Schritt drei der Installation ausgeführt werden.

8.2.1 Installation PHPMyAdmin

Schritt 1:

Um das PHPMyAdmin-Paket auf dem Raspberry Pi zu installieren, müssen wir den folgenden Befehl ausführen:

```
sudo apt install phpmyadmin
```

PHPMyAdmin beginnt nun mit der Installation. Als Typ des Webservers soll dabei „apache2“ ausgewählt werden.

Schritt 2:

Als Nächstes muss PHPMyAdmin konfiguriert werden, um eine Verbindung zu dem MYSQL-Server herzustellen.

Die nächste Eingabeaufforderung muss mit "**<Yes>**" ausgewählt werden:

Um den Server zu sichern, muss anschließend ein Passwort ausgewählt werden.

Dieses Passwort wird verwendet, um eine Verbindung zum MySQL-Server herzustellen.

Schritt 3:

Mit dem abgeschlossenen PHPMyAdmin-Installationsprozess gibt es noch eine letzte Sache. PHPMyAdmin wird standardmäßig verbieten, sich mit dem "root" Benutzer bei der PHPMyAdmin-Schnittstelle anzumelden.

Stattdessen muss ein neuer Benutzer erstellt werden, sodass auf PHPMyAdmin zugegriffen werden kann.

Um dies zu tun, muss sich der "root" Benutzer bei der MySQL-Befehlszeilenschnittstelle anmelden:

```
sudo mysql -u root -p
```

Nun muss der folgende Befehl ausgeführt werden, um einen Benutzer zu erstellen und ihm den Zugriff auf alle Datenbanken auf dem MySQL-Server zu ermöglichen. „username“ und „password“ sollen dabei durch den jeweiligen Benutzernamen und das Passwort ersetzt werden.

```
GRANT ALL PRIVILEGES ON *.* TO 'username'@'localhost' IDENTIFIED BY  
'password' WITH GRANT OPTION;
```

Um die MySQL-Befehlszeilenschnittstelle zu verlassen, kann im Terminal "quit" eingeben werden.

Schritt 4:

Daraufhin muss Apache neu gestartet werden, damit die Änderungen wirksam werden:

```
sudo systemctl restart apache2
```

Schritt 5:

Bei einer erfolgreichen Installation müsste die PHPMyAdmin-Oberfläche nun mit einem Webbrowser erreichbar sein (siehe Abbildung 29). Dazu muss „<http://localhost/phpmyadmin>“ in die Suchleiste eingegeben werden.¹

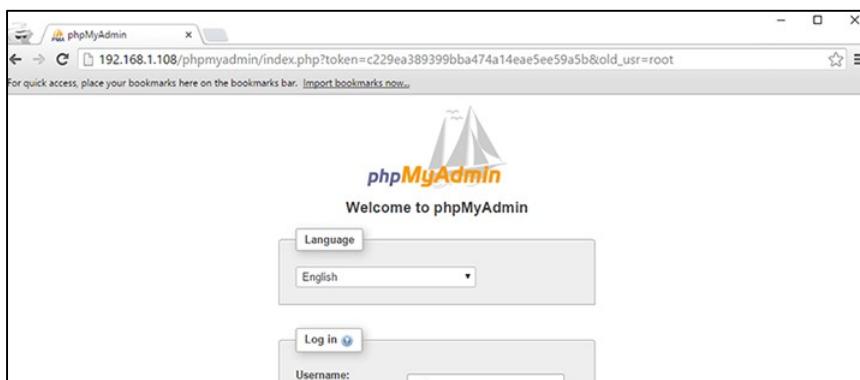


Abbildung 29: PHPMyAdmin-Oberfläche

¹ Vgl. (o.V., pimylifeup.com, 2019)

8.3 Verwendung im Projekt

In diesem Projekt wird „MariaDB“ – ein Fork der relationalen Datenbank MySQL verwendet, welche die Eigenschaften des ursprünglich entwickelten Datenbank-Managementsystems in sich vereint.

Die MariaDB- Datenbank wird in diesem Projekt dazu verwendet, um das PHP-Skript auf dem Apache- Server mit dem Java Programm zu verbinden. In ihr werden in verschiedenen Tabellen die Daten gespeichert, die auf der Steuerungsseite vom Benutzer verändert werden können. Diese Daten können anschließend vom Java Programm über einen SQL- Datenbankaufruf eingelesen werden. Die Datenbank bildet also die Schnittstelle zwischen der PHP- Steuerungsseite und dem Java Steuerungsprogramm (siehe Abbildung 30).

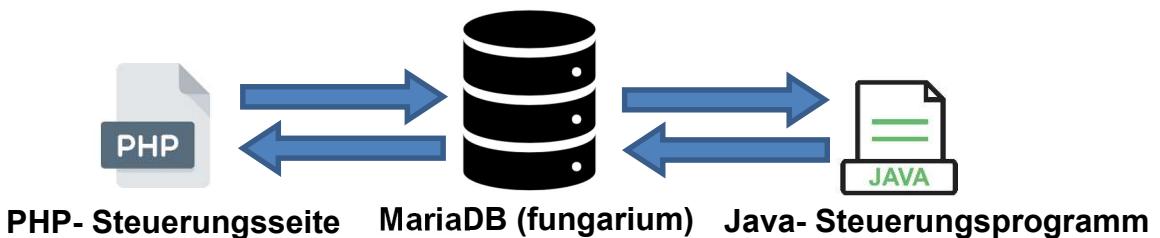


Abbildung 30: Schaubild PHP/ MariaDB/ Java

Für das „Fungarium“-Projekt soll nun im Folgenden eine eigene Datenbank angelegt werden. Der Name der Datenbank in diesem Projekt ist „fungarium“.

8.4 Datenbank erstellen

Dies kann zum einen über die zuvor installierte Verwaltungsoberfläche PHPMyAdmin oder über das Terminal gemacht werden.

Über das Terminal kann die benötigte Datenbank „fungarium“ mit dem „root“ Benutzer mit folgendem Befehl erstellt werden.

```
mysql -u root -e "create database fungarium;"
```

Der „root“ Benutzer hat kein Passwort.

8.5 Datenbanktabellen

Das Projekt benötigt fünf Tabellen, die in der Datenbank „fungarium“ gespeichert sind. Diese Tabellen werden verwendet, um verschiedene Daten des Projekts zwischenzuspeichern und das Java-Steuerungsprogramm mit der PHP-Steuerungsseite zu verbinden.

Die Bezeichnung der verwendeten Tabellen kann aus Abbildung 31 entnommen werden. Wichtig dabei ist, dass die Tabellen nicht manuell vom Entwickler, sondern **automatisch beim ersten Programmdurchlauf** des Java -und PHP-Programms erstellt werden. Deswegen handelt es sich bei der folgenden Beschreibung um einen reinen Überblick der verwendeten Datenbankstruktur.

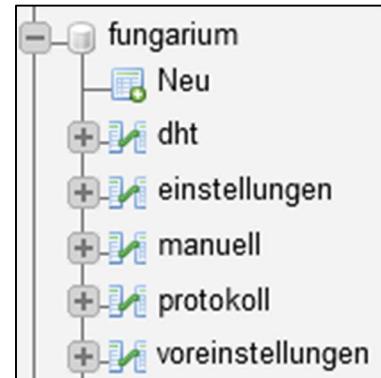


Abbildung 31: Tabellen in DB

8.5.1 Tabelle Einstellungen

Die Tabelle „**einstellungen**“ wird für die Licht, Temperatur, Feuchtigkeit und Lüftungseinstellungen verwendet. Diese Daten können auf der PHP- Startseite eingestellt werden.

Die Tabelle besteht aus vier Spalten (siehe Abbildung 32).

ID	Bezeichnung	Wert	Zugriff
1	Lichtstart	16:02	2021-01-19 18:39:45
2	Lichtende	16:06	2021-01-19 18:39:45
3	Mindesttemperatur	22	2021-01-19 18:39:45
4	Höchsttemperatur	33	2021-01-19 18:39:45
5	MindesttemperaturNacht	22	2021-01-19 18:39:45
6	HöchsttemperaturNacht	33	2021-01-19 18:39:49
7	Mindestfeuchtigkeit	99	2021-01-19 18:39:54
8	Befeuchtungsdauer	5	2021-01-19 18:40:01
9	Lüftungsdauer	6	2021-01-19 18:40:06
10	Lüftungsintervall	255	2021-01-19 18:40:20

Abbildung 32: Tabelle *einstellungen*

Die Spalte „**ID**“ gibt dabei dem jeweiligen Datensatz eine eindeutige Zuweisung mit einem Integer. Das Attribut „**Bezeichnung**“ und „**Wert**“ beschreiben um welche Einstellung es sich im jeweiligen Datensatz handelt und welchen Wert der Datensatz aufweist. Mit „**Zugriff**“ wird ein Zeitstempel gespeichert, der angibt, wann der Datensatz zuletzt aktualisiert wurde.

8.5.2 Tabelle Protokoll

Die Tabelle „**protokoll**“ wird für das Protokoll, welches auf der Steuerungsseite angezeigt wird, verwendet. Dabei wird auf der Steuerungsseite sowohl die Tabelle als auch die Diagramme zu Licht, Heizung, Kühlung, etc. aus „**protokoll**“ gebildet. Die Tabelle besteht aus fünf Spalten (siehe Abbildung 33).

ID	Bauteil	An	Eintrag	Zugriff
151	NULL	NULL	Sensordaten: [Fungarium, 42.0% , 19.9°C]	2021-04-08 17:35:52
152	NULL	NULL	Sensordaten: [Fungarium, 63.1% , 19.8°C]	2021-04-08 17:37:53
153	fogger	0	Switch off Actor fogger	2021-04-08 17:38:51
154	lueftung	0	Switch off Actor lueftung	2021-04-08 17:38:52
155	NULL	NULL	Ends Humidification routine	2021-04-08 17:38:53
156	NULL	NULL	Start Humidification routine	2021-04-08 17:39:03
157	lueftung	1	Switch on Actor lueftung	2021-04-08 17:39:04
158	fogger	1	Switch on Actor fogger	2021-04-08 17:39:07
159	fogger	0	Switch off Actor fogger	2021-04-08 17:39:18

Abbildung 33: Tabelle protokoll

Das Attribut „**Bauteil**“ gibt an welches Bauteil im jeweiligen Datensatz verwendet ist. Daraus können auf der Steuerseite, in Verbindung mit dem Attribut „**An**“, die einzelnen Diagramme zu den jeweiligen Bauteilen einfacher generiert werden. Wenn jedoch „**Einträge**“ vorhanden sind, die wie bei ID „3“ keine Verbindung mit einem Bauteil aufweisen, dann werden deren Attributen „**Bauteil**“ und „**An**“ auf NULL gesetzt.

8.5.3 Tabelle DHT

Für die Temperatur und Luftfeuchtigkeit, die vom DHT Sensor gemessen werden, gibt es in der Datenbank die Tabelle „dht“. In dieser wird die Temperatur in Grad Celsius und die relative Luftfeuchtigkeit in Prozent zum Messzeitpunkt, gespeichert (siehe Beispiel in Abbildung 34).

ID	Temperatur	Luftfeuchtigkeit	Zugriff
1	22	66	2021-01-04 18:44:42
2	24	77	2021-01-04 20:17:21
3	28	99	2021-01-04 20:17:48

Abbildung 34: Tabelle dht

Die Datensätze der Tabelle werden für die Temperatur -und Feuchtigkeitsdarstellung verwendet. Dabei werden die Daten jeweils als Graph und in einer Tabelle mit äquivalenten Werten dargestellt.

8.5.4 Tabelle Manuell

In der Datenbanktabelle „manuell“ werden die manuellen Einstellungen angezeigt, welche mit der Checkbox „Auto/Manuell“ auf der „Menü“- Seite angezeigt werden (siehe Abbildung 35).

ID	Bezeichnung	Wert	Zugriff
1	licht	1	2021-05-08 18:50:30
2	heizung	0	2021-05-08 18:50:29
3	kuehlung	0	2021-05-08 18:50:29
4	fogger	0	2021-05-08 18:50:28
5	lueftung	1	2021-05-08 18:50:28
6	lueftungNiedertourig	0	2021-05-08 18:50:27
7	sensorneustart	1	2021-05-08 18:50:32
8	befeuchtungsroutine	1	2021-05-08 18:50:35
9	autoManu	1	2021-05-08 18:50:19

Abbildung 35: Tabelle manuell

Der Wert der „Auto/Manu“- Checkbox wird dabei in dieser Tabelle unter der Bezeichnung „autoManu“ gespeichert. Die Spalte „Wert“ ist vom Typ TINYINT und kann deswegen nur eine „0“, „1“ oder „NULL“ (In diesem Fall nicht zulässig, da eine Checkbox nur den Wert „0“ oder „1“ annehmen kann) enthalten.

Der Wert „0“ bedeutet, dass sich das System im automatischen Modus befindet. Mit der Betätigung der Checkbox wird der Wert „1“ in der Tabelle vermerkt und das System befindet sich im manuellen Modus. Darin können die zuvor ausgewählten Komponenten vom Anwender auf der Steuerungsseite manuell ein- und ausschalten werden.

In ID zwei bis sechs werden die Werte der Komponenten-Checkboxen gespeichert, die der Anwender manuell ein- und ausschalten kann.

Mit den letzten beiden Einträgen kann ein Signal zum Steuerungsprogramm gesendet werden, sodass ein Sensorneustart oder ein Start der Befeuchtungsroutine erfolgt. Die Betätigung des jeweiligen Buttons auf der Steuerungsseite schreibt damit eine „1“ in die Datenbanktabelle. Dieser Eintrag wird bei der Ausführung der Aktion im Steuerungsprogramm auf „0“ zurückgesetzt, sodass kein doppeltes Ausführen entstehen können.

8.5.5 Tabelle Voreinstellungen

In der Tabelle „**voreinstellungen**“ werden die Voreinstellungen angezeigt, die vom Anwender auf der Steuerungsseite mit der Checkbox „Voreinstellungen“ eingeblendet werden können. Darunter fallen ebenso die eingegebenen Pinbelegung, die vom Anwender auszuwählen sind.

ID	Bezeichnung	Wert	Pin	Zugriff
1	licht	1	2	2021-05-06 21:19:25
2	heizung	0	4	2021-05-06 21:19:25
3	kuehlung	0	6	2021-05-06 21:19:26
4	fogger	1	10	2021-05-06 21:19:28
5	lueftung	1	9	2021-05-06 21:19:27
6	lueftungNiedertourig	0	8	2021-05-06 21:19:31
7	sensor	1	17	2021-05-06 21:19:30
8	terraFunga	0	NULL	2021-05-06 21:27:08
9	LichtAlsHeizung	1	NULL	2021-05-08 18:44:36

Abbildung 36: Tabelle Voreinstellungen

In ID eins bis fünf können die vorhandenen Komponenten eingestellt werden. Zu den Voreinstellungen gehört ebenfalls die Auswahl „Licht als Heizung“, durch welche das Licht bei fehlender Heizung deren Aufgabe übernehmen kann. „TerraFunga“ beschreibt den Zustand der Checkbox „Terrarium/Fungarium“.

9 PHP- Steuerungsseite

Die PHP-Steuerungsseite beinhaltet verschiedene PHP-Dateien, die für die Struktur, das Erscheinungsbild und die Funktion der Seite verantwortlich sind. Die Seite und die Verwendung der verschiedenen Programmiersprachen sollen in diesem Kapitel näher erläutert werden.

9.1 Programmiersprachen

Der Programmcode der Steuerungsseite enthält die Sprachen HTML, CSS, PHP, JavaScript und SQL.

Mit **CSS** werden Gestaltungsanweisungen erstellt, die vor allem im Zusammenspiel mit HTML eingesetzt werden. **HTML** legt die Struktur der Steuerungsseite fest und CSS ist dabei für das optische Erscheinungsbild verantwortlich. Die externen CSS-Stylesheets, die in diesem Projekt verwendet werden, sind im Ordner „css“ abgelegt.

Mit der Sprache **PHP** werden die Einstellungsdatensätze der Seite „index.php“ über „form_submit“ in die verwendete Datenbank geladen. Auf der anderen Seite werden die DB-Daten des DHT-Sensors und des Protokolls aus der DB geladen, um sie auf der Steuerungsseite in einem Diagramm und einer Tabelle zu visualisieren. Dazu hilft die Datenbanksprache **SQL**, um in Verbindung mit PHP die benötigten Anfragen an die DB ausführen zu können.

JavaScript ist in diesem Projekt für die Darstellung der Diagramme und Tabellen auf den Seiten „temperatur.php“, „luftfeuchtigkeit.php“ und „protokoll.php“ verantwortlich. Die Diagramme werden mit Hilfe des Moduls „chart.js“ erstellt. Die Tabellen wiederum verwenden das Modul „datatables.js“.

9.2 Struktur des Programmcodes

Die Struktur des Programmcodes der Steuerungsseite ist im folgenden Schaubild

(**Fehler! Verweisquelle konnte nicht gefunden werden.**) verdeutlicht:

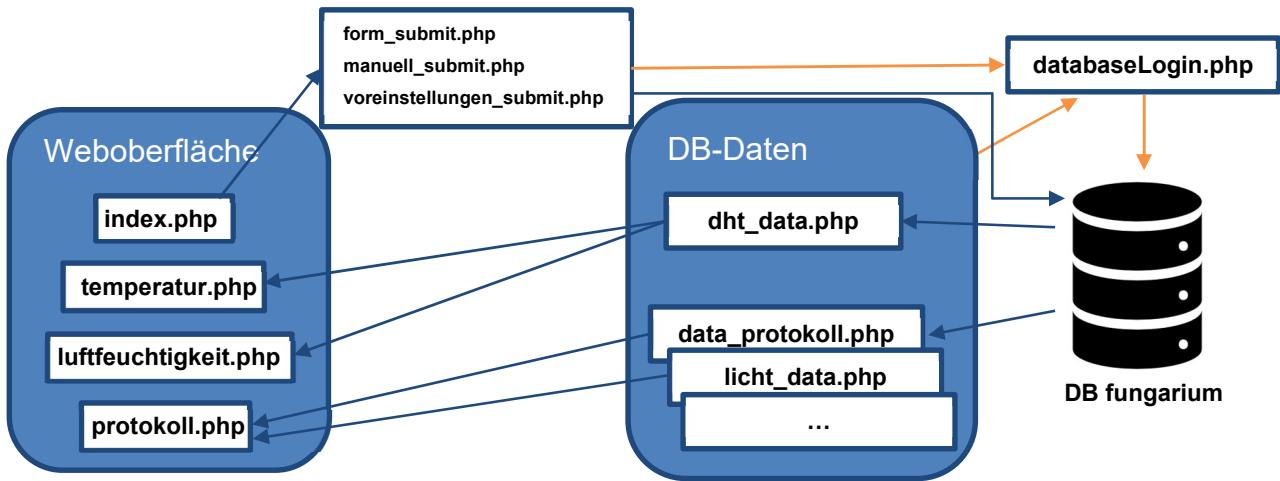


Abbildung 37: Schaubild Programmstruktur PHP

Die blauen Pfeile des Schaubildes beschreiben den Datenfluss der Einstellungs-, Sensor -und Protokolldaten. Die orangenen Pfeile stellen die Verbindung der Skripte zu „**databaseLogin.php**“ dar. Dabei wird in den Skripten, die sich unter „DB-Daten“ befinden und in „**form_submit.php**“, „**manuell_submit.php**“ und „**voreinstellungen_submit.php**“ der PHP-Code „**databaseLogin.php**“ aufgerufen, um eine Verbindung mit der Datenbank „fungarium“ herzustellen. Diese Verbindung ist essenziell, um Datensätze aus der Datenbank zu lesen oder sie in die DB zu schreiben.

Dabei ist wichtig, dass der Entwickler im Skript „**databaseLogin.php**“ seinen **Benutzernamen und Passwort** anpasst, sodass eine fehlerfreie Anmeldung gewährleistet ist.

Die Weboberfläche besteht aus vier PHP- Dateien. Diese Dateien bilden die Oberfläche des Apache-Servers, die vom Nutzer sichtbar sind. Die Startseite des Webservers bildet die Seite „`index.php`“. Die Seite ermöglicht dem Benutzer verschiedene Einstellungen, wie bspw. Lichtstart, Mindesttemperatur, etc. zu konfigurieren. Über ein Dropdown-Menü können die anderen drei Seiten erreicht werden, auf welchen die Temperatur und Luftfeuchtigkeit des DHT-Sensors und die Protokoll-Daten des Java-Programms eingesehen werden können.

Die Temperatur -und Luftfeuchtigkeitsdaten werden in „**temperatur.php**“ und „**luftfeuchtigkeit.php**“ in Form eines Liniendiagramms und einer Tabelle dargestellt.

Die Protokolldaten des Java- Steuerungsprogramms werden in einer Tabelle dargestellt. Um die Überwachung der im Projekt benutzten Bauteile, wie bspw. Licht oder Heizung, benutzerfreundlicher zu gestalten, sind auf der Seite „**protokoll.php**“ digitale Diagramme zu jedem verwendeten Bauteil gegeben. Anhand der Protokolldaten werden die Zustandsübergänge in den Diagrammen dargestellt.

Die Einstellungsdaten, die in der Startseite des Webservers „index.php“ eingestellt werden können, werden über ein Formular an die Datenbank „fungarium“ gesendet. Dies geschieht bei Betätigung des Buttons: „Submit/Daten absenden“ über die PHP-Seite „**form_submit.php**“. Diese Datensätze werden in der Datenbanktabelle „**einstellungen**“ gespeichert und können vom Java-Steuerungsprogramm zur Verwendung aufgerufen werden.

Die Voreinstellungen, welche die vorhandenen Komponenten und die Pinbelegung, wie auch die Auswahl für die Funktion „Licht als Heizung“ und die Auswahl des Systems (Fungarium/Terrarium) beinhalten, werden über das Skript „**voreinstellungen_submit.php**“ an die DB übermittelt. Die manuellen Daten (manueller Aktorbetrieb, manuelle Aktionen) benutzen das Skript „**manuell_submit.php**“, um die Daten an die DB zu übergeben.

Die DHT-Sensordaten, die in der Datenbanktabelle „**dht**“ gespeichert sind, werden vom Skript „**dht_data.php**“ abgerufen. „dht_data.php“ stellt eine JSON-Datei mit den jeweiligen Datensätzen bereit, auf welche die Seiten „temperatur.php“ und „luftfeuchtigkeit.php“ zugreifen. Unter der Verwendung der JavaScript-Module „chart.js“ und „datatables.js“ werden die Temperatur- und Feuchtedaten letztendlich auf den jeweiligen Seiten visualisiert. Die Mindest -und Höchsttemperatur aus der DB-Tabelle „einstellungen“ werden der JSON-Datei hinzugefügt. Analog dazu stellt das Skript „**data_protokoll.php**“ eine JSON-Datei mit den Datensätzen aus der Tabelle „**protokoll**“ bereit. Diese Daten füllen die Tabelle auf der Seite „protokoll.php“.

Um die Zustandsdiagramme der Bauteile auf der Protokollseite darzustellen, müssen die Daten der Protokolltabelle modifiziert werden. Diese Modifikation wird in den jeweiligen Skripten, wie bspw. „licht_data.php“, „heizung_data.php“, etc. durchgeführt. Anschließend wird eine JSON-Datei bereitgestellt, aus welcher die Daten für die jeweiligen Diagramme erstellt werden können.

Zum Test und zur visualisierten Darstellung der PHP-Dateien kann der Apache-Server unter der URL: „**fungarium.ddnss.de**“ aufgerufen werden. Da der Server mit einer „.htaccess“ -Datei abgesichert ist, wurde folgender Benutzer zu Testzwecken angelegt:

- **Nutzername:** tadeusz
- **Passwort:** 1212

10 Java-Steuerprogramm

Bei der Sprache Java handelt es sich um eine objektorientierte Programmiersprache, welche von der Firma Oracle verwaltet wird. Die Programmiersprache gehört zu der Java-Technologie, die sich weiter aus dem Java-Entwicklerwerkzeug und der Java-Laufzeitumgebung zusammensetzt.

Das in dieser Projektarbeit verfasste Java-Steuerprogramm wird verwendet, um das Fungarium zu automatisieren. In diesem Kapitel soll dabei die Struktur des Programms genauer beleuchtet werden. Des Weiteren befinden sich in diesem Abschnitt Erklärungen zu Methoden, welche die Implementierung von externen Bibliotheken erfordern. Für ein genaueres Verständnis bezüglich der Funktionen und dem Ablauf des Programms wird an dieser Stelle auf das Kapitel 2.1.3 Steuerprogramm verwiesen.

10.1 Installation

Die in diesem Projekt verwendete Java-Version ist die OpenJDK 8. Um OpenJDK 8 auf dem Raspberry Pi zu installieren, wird folgender Befehl ins Terminalfenster eingegeben:

```
sudo apt install open-8-jdk
```

Sollten mehrere Java-Versionen auf dem Raspberry installiert sein, so ist vor dem Start des Programms zu überprüfen, dass die richtige Version als Standard-Java-Version festgelegt ist. Die Standard-Java-Version kann über die Eingabe von folgendem Befehl eingesehen werden:

```
java -version
```

Um eine andere Java-Version festlegen zu können, wird zuerst folgender Code in den Terminal eingegeben:

```
sudo update-alternatives --config java
```

Anschließend erscheint eine Liste mit den installierten Java Versionen (Vgl. Abbildung 38). Die Standardversion ist über ein „*“-Symbol gekennzeichnet.

```
pi@raspberrypi:~ $ sudo update-alternatives --config java
There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                Priority      Status
-----+
  0            /usr/lib/jvm/java-11-openjdk-armhf/bin/java      1111      auto mode
  1            /usr/lib/jvm/java-11-openjdk-armhf/bin/java      1111      manual
* 2            /usr/lib/jvm/java-8-openjdk-armhf/jre/bin/java    1081      manual

Press <enter> to keep the current choice[*], or type selection number: █
```

Abbildung 38: Auswahl Java-Version

In der dargestellten Abbildung ist bereits die richtige Version ausgewählt.

Bei falsch ausgewählter Version muss die entsprechende Auswahlnummer eingegeben werden. Dies Auswahl ist mit betätigen von Enter zu bestätigen.

10.2 Struktur des Programmcodes

Die Struktur des Programmcodes des Java-Steuerprogramms ist in folgendem Organigramm (Abbildung 39) verdeutlicht.

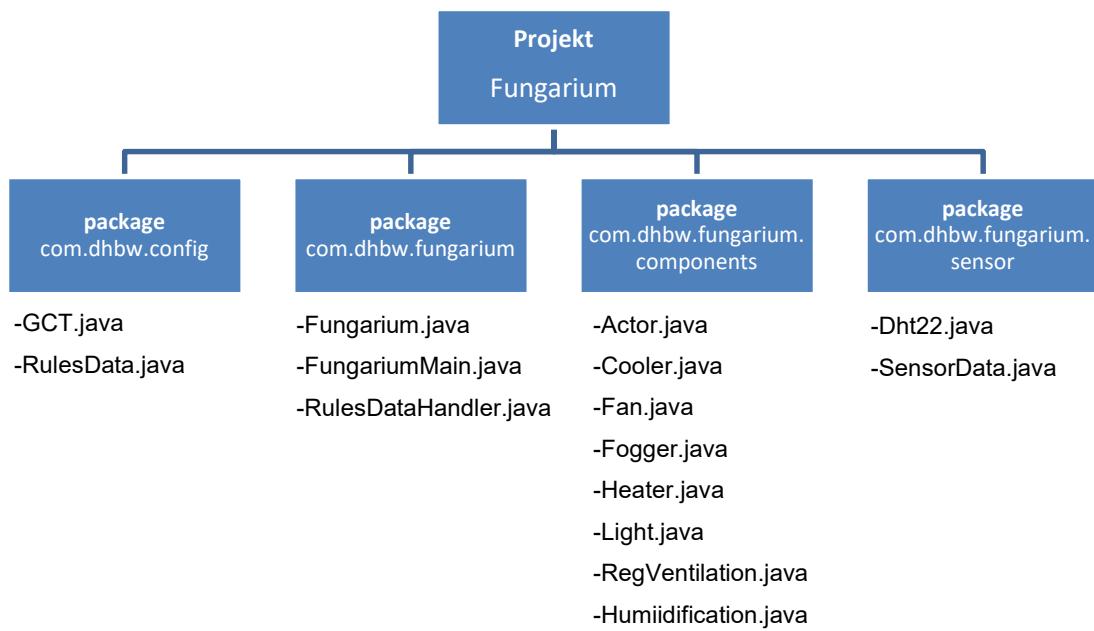


Abbildung 39: Programmstruktur Java

In der Abbildung ist zu sehen, dass sich das Projekt Fungarium in insgesamt 4 packages aufteilt. In den jeweiligen packages befinden sich wiederum die zugehörigen Java-Klassen. Diese Struktur ist auch beim Öffnen des Projekts mit Hilfe der Eclipse IDE gegeben.

Im package „**com.dhbw.config**“ befindet sich die Klassen „GCT“ und „RulesData“. Bei der Klasse „GCT“ (Global Control Table) handelt es sich um eine „Singelton class“. Dies bedeutet, dass von dieser Klasse lediglich ein Objekt instanziert werden kann. Das „Singelton pattern“ ist ein beliebtes design pattern, welches in der Java-Programmierung häufig zum Einsatz kommt.¹

Im Projekt wird dieses Design verwendet, um Werte abzuspeichern, welche in mehreren Klassen abrufbar sein müssen, jedoch in jeder Klasse den gleichen Wert einnehmen. Dazu gehören beispielsweise Zustandsvariablen (z.B.

Befeuchtungsroutine läuft oder nicht→ vom Typ boolean) und die Einstellparameter, welche über den Anwender vorgegeben sind.

Die Klasse „RulesData“ umfasst wiederum alle Einstellparameter als Attribute. Aufgrund dessen gibt es auch in der Klasse „GCT“ ein Objekt des Typs RulesData. Das Objekt wird bei jedem Lesevorgang von der Datenbank auf die aktuellen Daten synchronisiert. Dadurch können die aktuellen Einstellparameter in jeder Klasse über den Global Control Table abgerufen werden.

In der Klasse „FungariumMain“ im package „**com.dhbw.fungarium**“ befindet sich die Main Methode. Die Main Methode dient in der Java Umgebung als Startpunkt zur Ausführung des Codes.

Im Falle des Steuerprogramms wird in der Main Methode lediglich die Methode „doFungarium“ der Klasse „Fungarium“ aufgerufen. Dieser Aufbau wurde gewählt, um vermeiden zu können, dass alle Methoden „static“ sind.

In der Methode „doFungarium“ spielt sich das zentrale Geschehen der Automatisierung ab. Zu Beginn der Methode wird die Verbindung zur Datenbank aufgestellt. Des Weiteren werden die Systemkonfigurationen von der Tabelle „voreinstellungen“ eingelesen. Dazu zählt beispielsweise die GPIO Pinbelegung. Im nachfolgenden Teil wird an einer Stelle des Programms gewartet, bis Sensordaten in die sogenannten „sensorDataQueue“ hinzugefügt werden (eine genauere Erläuterung dazu folgt später). Diese Daten werden im ersten Schritt auf deren Plausibilität geprüft. Bei erfolgreicher Prüfung werden die Daten in die Datenbank geschrieben. Befindet sich das System im Automatikbetrieb, so wird anschließend überprüft, ob eine der Hardwarekomponenten zur Manipulation der Umweltbedingungen eingeschalten oder ausgeschalten werden muss. Im Falle von

¹ Vgl. (tutorialspoint, 2020)

einer fehlerbehafteten Messung wird ein Neustart des Sensors eingeleitet. Nach drei fehlerhaften Messungen in Folge wird das System in Notbetrieb versetzt. Diese Routine wird immer beim Empfang von Sensordaten durchgeführt.

Die Klasse „RulesDataHandler“ beschreibt dagegen das Einlesen der Einstellparameter aus der Datenbank.

Unter dem package „**com.dhbw.fungarium.components**“ ist unter anderem die Klasse „Actor“ zu finden. „Actor“ ist eine Oberklasse, die die Methoden „on“ und „off“ beinhaltet, welche zum Schalten der GPIO-Pins verwendet werden. Die Klassen „Light“, „Heater“, „Fogger“, „Fan“ und „Cooler“ sind dabei von der Klasse „Actor“ abgeleitet. Jede dieser Klassen muss dabei einen definierten GPIO-Pin besitzen. Der Vorteil hierbei ist, dass die Methoden „on“ und „off“ der „Actor“ -Klasse dem jeweiligen Objekt vererbt werden. Zusätzlich befinden sich in der Klasse „Actor“ zwei Methoden, welche zum Aktualisieren der Tabelle „manuell“ und „protokoll“ verwendet werden.

Die Klasse „RegVentilation“ beschreibt die regelmäßige Lüftung und ist von der Java Standardklasse „Thread“ abgeleitet. Dabei wird in der vererbten Methode „run“ unter gegebenen Voraussetzungen (z.B. Abfrage ob die Ventilationsroutine bereits läuft) die Methode „FanIn“ (=Lüftungsroutine) der Klasse „Fan“ aufgerufen. Anschließend wird über ein Gesamtintervall, welches sich zusammensetzt aus Belüftungsdauer und Lüftungsintervall, abgewartet. Dieser Ablauf wiederholt sich im Anschluss. Die Klasse „Humidification“ beschreibt dagegen die Befeuchtungsroutine. Im Falle der Befeuchtungsroutine werden zu Beginn der Routine die Lüftung niedertourig und der Fogger gestartet. Ist keine niedertourige Lüftung vorhanden, so wird der normale Lüfter angesteuert. Anschließend wird über die Befeuchtungsdauer abgewartet. Zum Abschluss werden beide Komponenten wieder ausgeschalten.

Im package „**com.dhbw.fungarium.sensor**“ befinden sich die Klassen „Dht22“ und „SensorData.“. Bei der Klasse „Dht22“ handelt es sich um eine abgeleitete Thread-Klasse, welche wiederum die Methode „run“ implementiert. Zusätzlich beinhaltet die Klasse die Methoden „sensorRestart“ (zum Neustarten des Sensors), „doMeter“ (durchführen einer Messung) und „readSensorData“ (liest die Daten vom DHT22 → genauere Beschreibung Abschnitt 0). Die Klasse „SensorData“ besitzt insgesamt vier

Attribute, welche die Temperatur, die Luftfeuchtigkeit, die Sensorposition und die aktuelle Zeit angeben.

Nachdem Einlesen der Sensordaten über den DHT22 werden diese in die „sensorDataQueue“ geschrieben. „sensorDataQueue“ ist eine Instanz der Klasse „ArrayBlockingQueue“. Diese Klasse enthält die Methode „take“, welche immer das oberste Objekt der Liste abruft und es anschließend entfernt. Ist kein Element in der Liste vorhanden so wartet das Programm an dieser Stelle bis ein Element verfügbar ist.¹

Die Methode wird im Hauptprogramm „Fungarium“ aufgerufen. Wird dementsprechend ein Element über den Sensor in die Queue geschrieben, werden anschließend die Checkroutinen (Plausibilitätscheck etc.) durchgeführt.

Mit Hilfe dieses Aufbaus können durch geringen Aufwand weitere Sensoren in das Programm zu implementieren.

¹ Vgl. (Oracle, 2020)

10.3 Logger (log4j)

Als „Logging“ bezeichnet man die automatische Erstellung eines Protokolls von Softwareprozessen. In diesem Projekt wird das Framework „log4j“ zum Loggen verwendet.

Das Framework „log4j“ lässt sich dabei auf vier verschiedene Arten vom Anwender konfigurieren:

- Über eine Konfigurationsdatei, die im XML-, JSON-, YAML- oder Properties Format verfasst ist.
- Durch Erstellen einer ConfigurationFactory- und Configuration-Implementierung im Programm.
- Programmgesteuert durch den Aufruf der in der Konfigurationsoberfläche bereitgestellten APIs. Dadurch können weiter Komponenten zur Standartkonfiguration hinzugefügt werden.
- Direkt im Programm durch das Aufrufen von Methoden der internen Logger-Klasse.¹

Im Projekt ist der Logger über eine XML-Datei konfiguriert. Dabei erfüllt er momentan folgende Funktionen:

- Ausgabe von Informationen: Auf der Console + in einem Log-File + Aktualisieren der Protokolltabelle der Datenbank. Im Falle der Protokolltabelle werden zwei verschiedene Logger verwendet. Beide Beschreibungen befinden sich in der Konfigurationsdatei. Einer der beiden Logger kann lediglich zum „loggen“ von Nachrichten verwendet werden wie beispielsweise „Wechsel in Notbetrieb“. Der andere Logger wird wiederum bei der Zustandsänderung eines Aktors eingesetzt.
- Speichern der Files: Pro Tag wird ein Log-File erstellt. Am Ende eines Tages wird das Log-File als zip-Datei in einen History Ordner verschoben. Die Log-Files sind in einem Unterordner mit dem Namen „logs“ innerhalb des Ordners „Fungarium“ zu finden. Der Hauptordner „Fungarium“ befindet sich wiederum im Ordner „pi“.

¹ Vgl. (The Apache Software Foundation, 2020)

10.4 Update gpio

Zur Ansteuerung der GPIO-Pins unter Java wird die Bibliothek pi4j verwendet. Da sich auf dem Raspberry Pi 4 die interne Verkabelung geringfügig verändert hat, ist meist eine Aktualisierung der Bibliothek WiringPi erforderlich. Bei Verwendung eines Raspberry Pi 4 ist deshalb sicherzustellen, dass die Version 2.52 des Dienstprogramms gpio verwendet wird.

Die aktuelle Version kann über den untenstehenden Befehl herausgefunden werden:

```
gpio -v
```

Ist eine veraltete Version vorhanden, so kann das Dienstprogramm über folgende Befehle aktualisiert werden:

```
cd /tmp  
wget https://project-downloads.drogon.net/wiringpi-latest.deb  
sudo dpkg -i wiringpi-latest.deb
```

10.5 Auslesen der Messwerte des DHT22 – Sensor

Um mit dem Raspberry Pi die Temperatur und die Luftfeuchtigkeit aufnehmen zu können wird ein DHT22 Temperatur- Luftfeuchtigkeitssensor verwendet. Aufgrund von Laufzeitproblemen ist es leider nicht möglich die Daten des Sensors mithilfe des Java-Programms auszulesen. Mehrerer Versuche dies mit Hilfe von Timing-Diagrammen zu realisieren sind erfolglos geblieben.

Abhilfe schafft in diesem Fall ein Python Script, welches an der entsprechenden Stelle im Java-Programm aufgerufen wird. Im Falle des Hobby-Fungariums wird das Python Skript in der Klasse „Dht22“ aufgerufen.

10.5.1 Adafruit

Zum Ansteuern des DHT22 Sensors wird die CircuitPython-DHT Library von Adafruit verwendet. Die Library kann dabei kostenlos mit folgendem Befehl auf das Raspberry Pi geladen werden:

```
pip3 install adafruit-circuitpython-dht  
sudo apt-get install libgpiod2
```

Der untenstehende in Python verfasste Code wird dabei verwendet, um die Luftfeuchtigkeit und die Temperatur über den DHT22 auszulesen.

Die ersten drei Zeilen des Codes beschreiben dabei die Einbindung der notwendigen Bibliotheken. In Zeile 5 wird festgelegt, an welchem GPIO Pin die Datenleitung des DHT22 angeschlossen ist. Im vorliegenden Fall wird GPIO4 zur Verbindung der Datenleitung verwendet. Bei Verwendung eines anderen Pins muss das Programm an dieser Stelle angepasst werden. Die darauffolgenden Zeilen lesen die Temperatur und Luftfeuchtigkeit. Nach der Aufnahme der Werte werden diese in das Terminal geschrieben. Das Terminal ist in diesem Fall die Schnittstelle zum Java-Programm. Im Java-Programm wird nach Aufruf des Python-Skripts die Konsole ausgelesen und dadurch die gemessenen Werte übernommen.

```
1 import time  
2 import board  
3 import adafruit_dht  
4  
5 dhtDevice = adafruit_dht.DHT22(board.D4)           GPIO4  
6  
7 try:  
8     temperature_c = dhtDevice.temperature  
9     humidity = dhtDevice.humidity  
10    print("{0:0.1f} {1:0.1f}“.format(temperature_c, humidity)  
11 except:  
12     print(error.args[0])
```

10.6 Herunterladen der Java-Applikation

Die angefertigte Java-Applikation sowie die notwendigen Bibliotheken sind via GitHub unter folgender URL verfügbar:

<https://github.com/svenSchelling/Fungarium>

GitHub ist ein netzbasierter Dienst zur Versionsverwaltung verschiedener Software-Projekte. Bevor der entsprechende Code über den Terminal heruntergeladen werden kann, muss im ersten Schritt „git“ auf dem Raspberry pi installiert werden. Dieser Schritt wird mit folgendem Befehl realisiert:

```
sudo apt install git
```

Nachdem git installiert wurde, kann anschließend die Java-Applikation und die entsprechenden Bibliotheken mithilfe folgender Codezeile heruntergeladen werden:

```
git clone https://github.com/svenSchelling/Fungarium.git
```

Nach Abschluss des Downloads ist unter dem Pfad „/home/pi“ ein Ordner mit dem Namen „Fungarium“ zu finden. Dieser Ordner weist folgende Architektur auf:

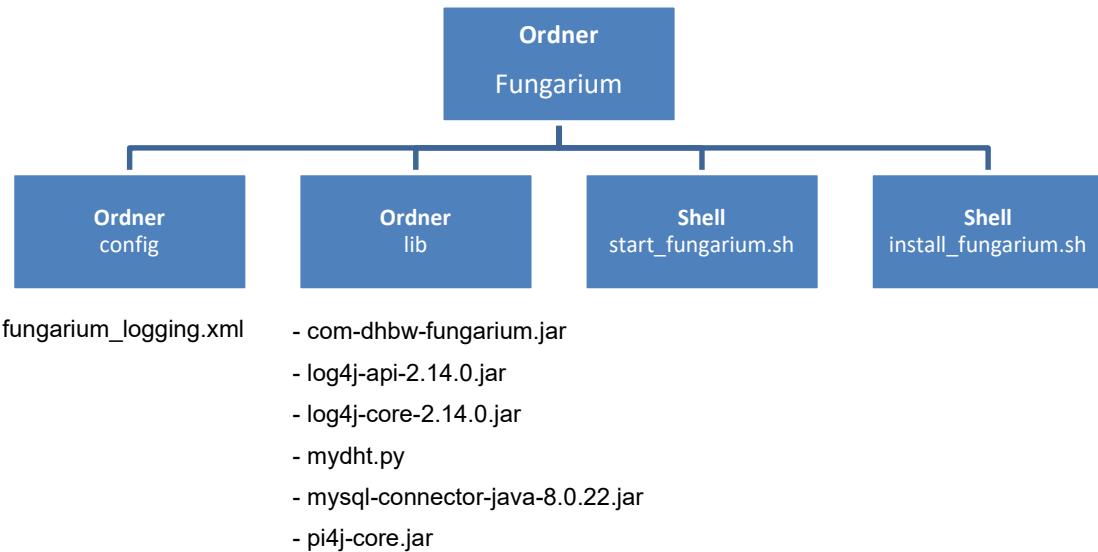


Abbildung 40: Struktur Fungarium Ordner

Direkt unter dem Ordner sind die Shell-Dateien „start_fungarium.sh“ und „install_fungarium.sh“ zu finden. Bei der erstgenannten Datei handelt es sich um eine Datei zum Starten der Java-Applikation. In dieser Datei wird unter anderem der Classpath deklariert. Des Weiteren werden hier zusätzlich Parameter deklariert (bspw. Username der DB), die dem Java Programm von außen übergeben werden.

Im Falle der zweiten Shell-Datei handelt es sich um das in Kapitel 5 beschriebene Bash-Skript, welches vom Anwender eingesetzt werden kann um die Installation zu automatisieren.

Im Ordner „config“ werden Konfigurationsdateien abgelegt. Im Falle des Fungariums befindet sich in diesem Unterordner lediglich eine XML-Datei, die zur Konfiguration des Loggers (log4j) verwendet wird.

Im „lib“ Ordner befinden sich die verschiedenen Bibliotheken, welche als JAR file abgelegt sind. Dazu zählt:

- „**pi4j-core.jar**“: Notwendig zur Ansteuerung der GPIO-Pins
- „**mydht.py**“: Python Skript zum Ansteuern des DHT22. An dieser Stelle ist nochmals anzumerken, dass in dieser Datei an der entsprechenden Stelle im Code die Nummer des GPIO-Pins angepasst werden muss, an dem die Datenleitung des DHT22 angeschlossen ist (siehe Absatz 10.5.1).
- „**com-dhbw-fungarium.jar**“: Beschreibung der eigentlichen Java-Applikation.
- „**log4j-api-2.14.0.jar**“ und „**log4j-core-2.14.0.jar**“: Zur Verwendung des Loggers.
- „**mysql-connector-java-8.0.22.jar**“: Bibliothek zur Verbindung mit der Datenbank.

Wie bereits erwähnt kann das Fungarium über die Shell-Datei „start_fungarium.sh“ gestartet werden. Dazu muss zuerst in den Ordner „Fungarium“ navigiert werden. Anschließend kann die Datei aufgerufen werden. Die entsprechenden Befehle sind im Folgenden abgebildet:

```
cd Fungarium  
./start_fungarium.sh
```

Meist ist der Start nicht direkt möglich, aufgrund der Berechtigung des jeweiligen Nutzers. Die Berechtigung kann über folgenden Befehl im Terminal verändert werden:

```
chmod 774 start_fungarium.sh
```

Anschließend kann die Applikation gestartet werden.

10.7 Einrichten eines Autostarts

Die Einrichtung eines Autostarts ist notwendig, wenn die Applikation nach dem Hochfahren des Raspberry Pi's automatisch starten soll. Dadurch muss das Programm nicht über die in Absatz 0 beschrieben Befehle manuell gestartet werden. Zum Einrichten eines Autostarts muss ein Desktop-file erstellt werden und unter dem Pfad „/etc/xdg/autostart/“ abgelegt werden. In diesem Desktop-file ist wiederum der Pfad zur Start-Datei anzugeben. Die jeweiligen Befehle lauten:

```
sudo nano /etc/xdg/autostart/fungarium.desktop
```

Mithilfe der Eingabe dieses Codes wird ein Desktop-file direkt im angesprochenen Ordner abgelegt. Dieses wird anschließend mit folgendem Inhalt gefüllt:

```
[Desktop Entry]
Name=fungarium
Exec=sh /home/pi/Fungarium/start_fungarium.sh
Terminal=false
```

Daraufhin ist der Autostart vollständig eingerichtet.

11 Hardware

Im folgenden Kapitel wird der Aufbau der verwendeten Aktoren (Lüfter, Licht, Fogger) und Sensoren (DHT22), wie auch die Verdrahtung und die Installation der einzelnen Komponenten, die im Projekt verwendet werden, beleuchtet. Schließlich werden optionale Komponenten beschrieben, die vom Anwender bei Belieben einfach und flexibel hinzugefügt werden können, um die Funktionalität des Fungariums/Terrariums zu erweitern.

11.1 Fungarium/Terrarium

Für das geschlossene System des Fungariums, liegt ein ausgesondertes Aquarium mit den Maßen 80x35x45cm vor (Abbildung 41).



Abbildung 41: Aquarium

Der Deckel des Aquariums ist eine laminierte Holzspanplatte. Er ist so konstruiert, dass er auf der oberen Kante des Aquariums aufliegt und an jeder Seite für 1,5 cm übersteht. Dies gewährleistet dem Anwender ein leichtes Anheben des Deckels. Am Deckel sind auf der Unterseite zwei Querstreben aus Holz angeschraubt, um die Position auf dem Aquarium zu fixieren. Für das Lüftungssystem sind an den gegenüberliegenden Ecken des Deckels Bohrungen angebracht.

11.2 Lüftungssystem

Das Lüftungssystem des Fungariums ist mit Hilfe eines Lüfters, eines Foggers, eines Eimers und der Verwendung von Schläuchen realisiert. Diese Komponenten sind in den folgenden Abbildungen dargestellt (Abbildung 42 und Abbildung 43).



Abbildung 42: Lüfter



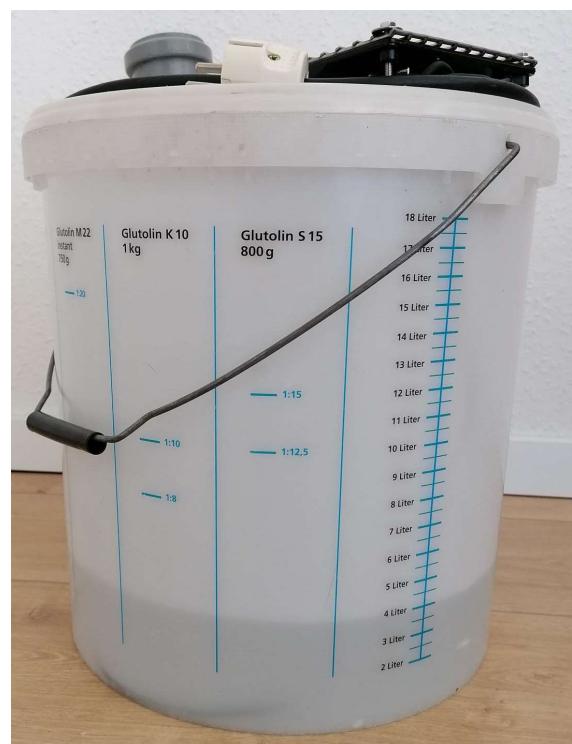
Abbildung 43: Fogger

Die Grundidee ist es die Luftfeuchtigkeit im Fungarium mit Hilfe des Lüftungssystems zu erhöhen und die Pilzzucht mit ausreichend Sauerstoff zu versorgen. Auf der anderen Seite ist es wichtig mit der Lüftung das Kohlenstoffdioxid aus dem System zu entfernen.

Der Zusammenbau des Lüftungssystems ist in den folgenden Abbildungen dargestellt.



Abbildung 44: Lüftungssystem



Der Eimer mit einem abnehmbaren Deckel, dient als Wasserspeicher, in welchem sich der Fogger befindet. Der Fogger ist dafür verantwortlich die Luftfeuchtigkeit im Inneren des Eimers zu erhöhen. Wird also die Befeuchtungsroutine gestartet fängt der Fogger an das Wasser im Eimer mit Ultraschalltechnik zu Nebel zu zerstäuben. Gleichzeitig wird der erzeugte Nebel mit Hilfe des startenden Lüfters über einen Schlauch in das Fungarium transportiert. Dadurch steigt die Luftfeuchtigkeit im Inneren des Fungariums an und das Kohlenstoffdioxid, welches sich am Boden des Systems befindet, wird aufgrund des entstehenden Überdrucks im System mit Hilfe eines Schlauchs an die Umwelt abtransportiert. Weitere Vorteile der Lüftung sind, dass die Substratfeuchte erhöht wird und Pilze oder Terrarien Tiere so vor dem Austrocknen geschützt werden.

Im Deckel des Eimers sind zwei Löcher gebohrt, um den Lüfter und die Rohr-Vorrichtung, welche mit dem Schlauch in das Fungarium führt, zu befestigen. Der Lüfter ist mit vier M4 Schrauben am Deckel befestigt (siehe Abbildung 46).



Abbildung 46: Befestigung Lüfter



Abbildung 45: Fogger im Eimer

Der Fogger befindet sich im Eimer, der mit Wasser gefüllt ist. Das Kabel des Foggers ist mit Heißkleber am Deckel des Eimers befestigt. Auf diese Weise kann der zweite Teil des Foggerkabels, an dem sich das Netzteil befindet, einfach in die Buchse eingesteckt werden.

Die Rohrvorrichtung ist durch einen Klemmmechanismus mit Hilfe einer Muffe an dem Deckel des Eimers befestigt. Diese einfache Vorrichtung erspart die Verwendung eines Klebers und dadurch ist die Rohrvorrichtung flexibel austauschbar (Abbildung 47).



Abbildung 47: Rohrvorrichtung

Um die Verbindung des Lüftungssystems (dem Eimer) mit dem Fungarium zu schaffen, wird der Schlauch in die Rohrvorrichtung eingeführt. Das andere Ende des Schlauchs wird in die Rohrvorrichtung des Fungariums geführt. Durch die Verwendung einer Dichtung wird ein optimaler Volumenstrom gewährleistet.

Die Idee der Rohrwanne, die sich innerhalb des Fungariums befindet, ist direkte Tropfen auf die Pilzzucht zu vermeiden, sodass das Kondensat aus der Wanne im Fungarium verdunsten kann. (siehe Abbildung 48).



Abbildung 48: Rohrwanne

Der Abluft Schlauch ist am Boden positioniert, damit durch die eingeblasene Luft, vorzugsweise das bodennahe Kohlenstoffdioxid, aus dem Fungarium abgeleitet werden kann. Dieser Schlauch ist mit Heißkleber am Deckel des Fungariums befestigt.



Abbildung 49: Schlauch zum Abtransport

Zur Ansteuerung des Lüfters wird ein Kabel mit Schutzleiter an dem Lüfter und dessen Gehäuse befestigt. Ein Schutzleiter muss verwendet werden, da der Lüfter ein metallisches Gehäuse aufweist. Der Lüfter kann mit Netzspannung von 230 Volt mit einer Frequenz von 50 Hertz betrieben werden.

Die Kabel sind zum Schutz mit einem Schrumpfschlauch ummantelt (siehe Abbildung 50).



Abbildung 50: Ansteuerung Lüfter

11.3 Licht

Das Licht des Fungariums ist im Zentrum des Deckels befestigt. Diese Befestigung erfolgt ebenfalls über eine Klemmung des Kabels der Lampenfassung (siehe Abbildung 51).



Abbildung 51: Licht

Zum Schutz der Lampe vor zu hoher Feuchtigkeit ist ein Marmeladenglas zur Hilfe genommen worden, welches über die Lampe aufgesetzt wird. Der Deckel des Marmeladenglases ist mit zwei Schrauben an der Unterseite des Deckels des Fungariums befestigt. So kann das Glas sehr einfach über die Lampe geschoben und geschlossen werden. Die Stromversorgung der Lampe erfolgt über die Verteilerbox. Dabei wird das Kabel durch den Deckel des Fungariums und ein Loch im Boden der Verteilerbox in diese geführt. Darin wird der Last -und Neutralleiter mit 230 Volt Netzspannung verbunden.

11.4 DHT22-Sensor

Der DHT22-Sensor (Abbildung 52) ist mit Hilfe eines beidseitigen Klebebandes an der Innenseite des auf dem Fungarium befindlichen Deckels angebracht. Dabei ist darauf zu achten, dass eine ausreichend große Distanz von Sensor zur Lichtquelle, sowie zur Luftzufuhr realisiert wird. Die Distanz ist maßgeblich für eine vertrauenswürdige Funktionsweise der Automatisierung. Werden zusätzlich Aktoren zur Kühlung und Heizung des Fungariums verwendet, so ist auch in diesem Fall auf eine bedachte Positionierung des Sensors hinzuweisen.

Der Sensor ist über einfache Jumper Kabel mit dem Raspberry Pi verbunden. Die Kabel sind dabei über eine durchgängige Bohrung von der Verteilerbox in das Fungarium geschleift. Im Falle der Kabellänge gilt die Regel: Umso kürzer umso besser. Da der Sensor bei zu groß gewählter Kabellänge Fehler aufweisen kann.



Abbildung 52: DHT22-Sensor

11.5 Verteilerbox mit Raspberry Pi

Die Grundidee der Verteilerbox ist es, eine Möglichkeit zu schaffen, die Komponenten einfach per Steckdose anschließen und steuern zu können. Dadurch kann ein aufwendiges Verdrahten der einzelnen Komponenten an das Relaisboard vermieden werden. Gleichzeitig ist es möglich sowohl das Relaisboard als auch das Raspberry Pi im Inneren der Verteilerbox zu verstauen, um somit zum einen die Verkabelungswege so gering wie möglich zu halten, als auch das Raspberry Pi vor möglichen Umwelteinflüssen zu schützen.

Die gesamte Box besteht aus einzelnen zugesägten Sperrholzplatten. Die Sperrholzplatten wurden nach dem Zusägen mit den notwendigen Bohrungen für die Steckdosen und Kabel Zu- und Abgänge versehen. Die Platten sind mit einem Holzleim zusammengeklebt.

Bei der Fertigung sowie dem Zusammenbau der Box wurden die in Anhang 13.4 abliegenden Fertigungszeichnungen als Vorlage verwendet. In Abbildung 53 ist auf der linken Seite die digitale Darstellung der Verteilerbox, auf der rechten Seite die eigentliche Box zu sehen. Die digitale Zeichnung wurde mit Hilfe der Applikation Shapr3D erstellt.

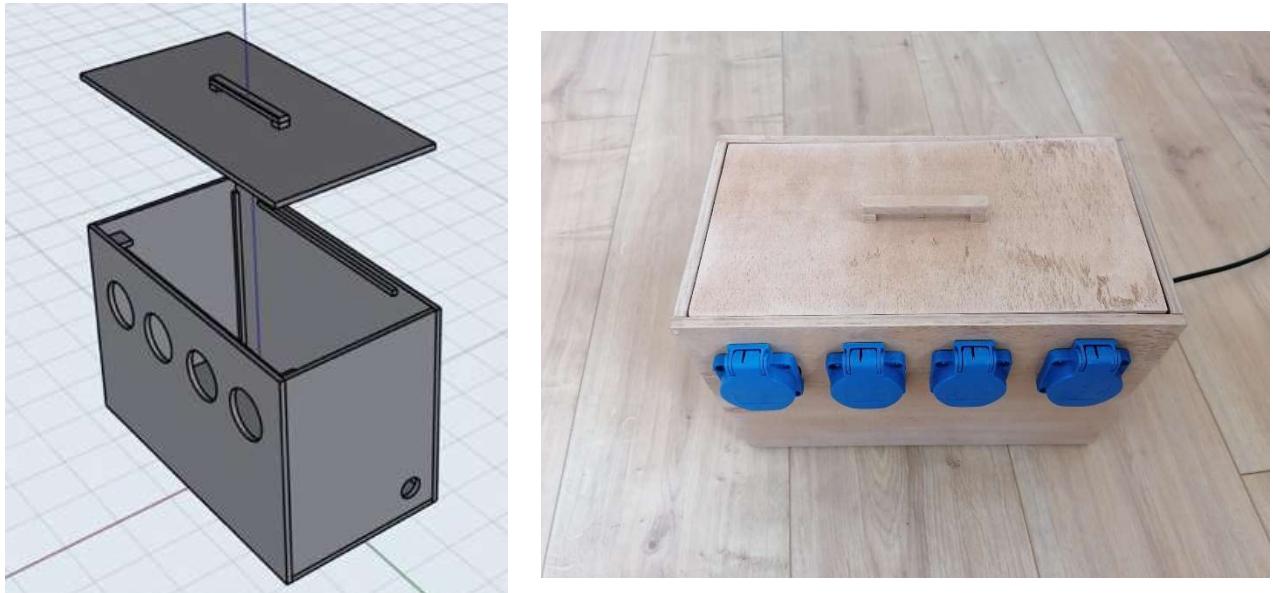


Abbildung 53: Verteilerbox

11.5.1 Seitenstabilisierung

Da die Sperrholzplatten eine geringe Dicke von 5 mm aufweisen, sind zur Stabilisierung der Klebeverbindungen in jeder Ecke der Box eine kleine Strebe eingeklebt. Eine der Seitenstreben ist in Abbildung 54 zu sehen. Des Weiteren sind eine Strebe an der Vorderwand sowie an der Rückwand der Verteilerbox mit einem Abstand von jeweils 5 mm zur Oberkante angebracht. Diese werden zur Ablage des Deckels der Verteilerbox verwendet, damit dieser nicht verrutschen kann.

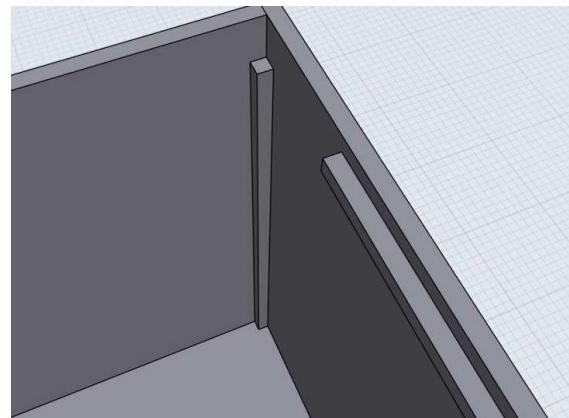


Abbildung 54: Seitenstreben

11.5.2 Elektrische Verkabelung

Die Strom- und Spannungsversorgung der Verteilerbox wird über einen zentralen SchuKo Stecker in die Box geschleift. Unmittelbar neben dem Anschluss des Hauptsteckers befindet sich die Verteilerdose auf der rechten Wandseite im Inneren der Box. Der Hauptstecker beziehungsweise die Zuleitung ist in Abbildung 55 dargestellt. Die Spannungsversorgung wird anschließend von der Verteilerdose an die Eingänge der Relais weitergeschleift.

Des Weiteren befindet sich innerhalb der Verteilerbox eine weitere Steckdose, welche eine Strom- und Spannungsversorgung bereitstellt, sofern der Hauptstecker eingesteckt ist. Über diese Steckdose ist wiederum das Raspberry Pi angeschlossen.



Abbildung 55: Hauptstecker/Zuleitung

Vom Relaisboard werden anschließend vier Ausgänge, in unserem Fall CH8-CH5 (Aufgrund von kürzeren Verkabelungswegen), über die Verteilerbox mit den Steckdosen verbunden. Dadurch ist es möglich die Steckdosen mithilfe der Relais CH8-CH5 zu schalten. Ein weiterer Ausgang des Relaisboards (CH4) wird dazu verwendet, um die Lichtquelle zu steuern. Neutralleiter sowie PE sind ebenfalls über die Verteilerbox mit den entsprechenden Komponenten verbunden. Die Verbindungen innerhalb der Verteilerdose sind mit Hilfe von Federklemmen realisiert.

In Abbildung 56 ist das Innenleben der Verteilerbox dargestellt. Dabei ist auf der rechten Seite die angesprochene Verteilerdose zu sehen. Unmittelbar am unteren Rand der Abbildung befinden sich dagegen die über das Relaisboard geschalteten Steckdosen. Das Relaisboard ist auf der gegenüberliegenden Seite mit Hilfe einer Schraubenverbindung an der Rückwand der Verteilerbox angebracht. Des Weiteren ist in der Darstellung die zusätzliche Steckdose im Inneren der Verteilerbox zu sehen, welche zur Stromversorgung des Raspberry Pi's dient.

In Anhang 13.5 befindet sich der zugehörige elektrische Schaltplan zur Verteilerbox.

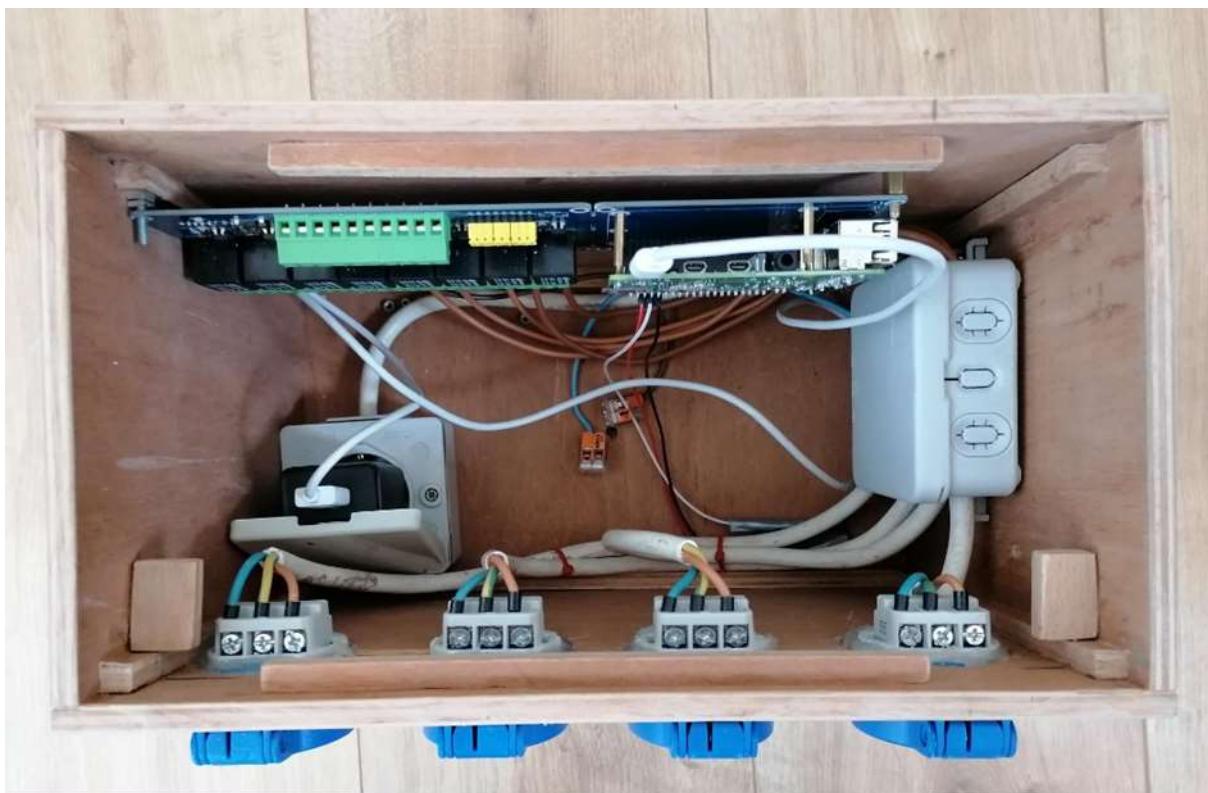


Abbildung 56: Innenleben Verteilerbox

11.5.3 Schutzabdeckung

Bei geöffnetem Deckel der Box sind speziell die offenen Kontakte an den Steckdosen sichtbar (Vgl. Abbildung 56). Um die Gefahr eines Stromschlages bei eingestecktem Hauptstecker weitestgehend zu vermeiden, wurde deshalb eine Schutzabdeckung konstruiert. In Abbildung 57 ist die Schutzabdeckung der Steckdosenkontakte abgebildet. Zur Halterung der Schutzabdeckung sind an beiden Außenwänden kleine rechteckige Sperrholzplatten angebracht. Einen zusätzlichen Halt bietet die Holzstrebe am oberen Rand der Vorderseite, welche zusätzlich zur Halterung des Deckels verwendet wird. Die Schutzabdeckung wird über das Einschieben

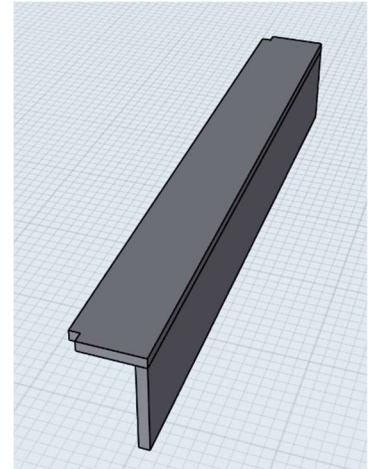


Abbildung 57: Schutzabdeckung

zwischen der Strebe und den seitlichen Holzplatten fixiert. Mit Hilfe Abbildung 58 soll die Verwendung der Schutzabdeckung nochmals verdeutlicht werden. Dabei sind die Stützen der Abdeckung rot markiert. Dazu zählen die bereits angesprochenen Seitenhalterungen sowie die Holzstrebe. Zusätzlich wird über den grünen Pfeil die Einschieberichtung markiert. Das untere Bild zeigt wiederum die fixierte Schutzabdeckung.

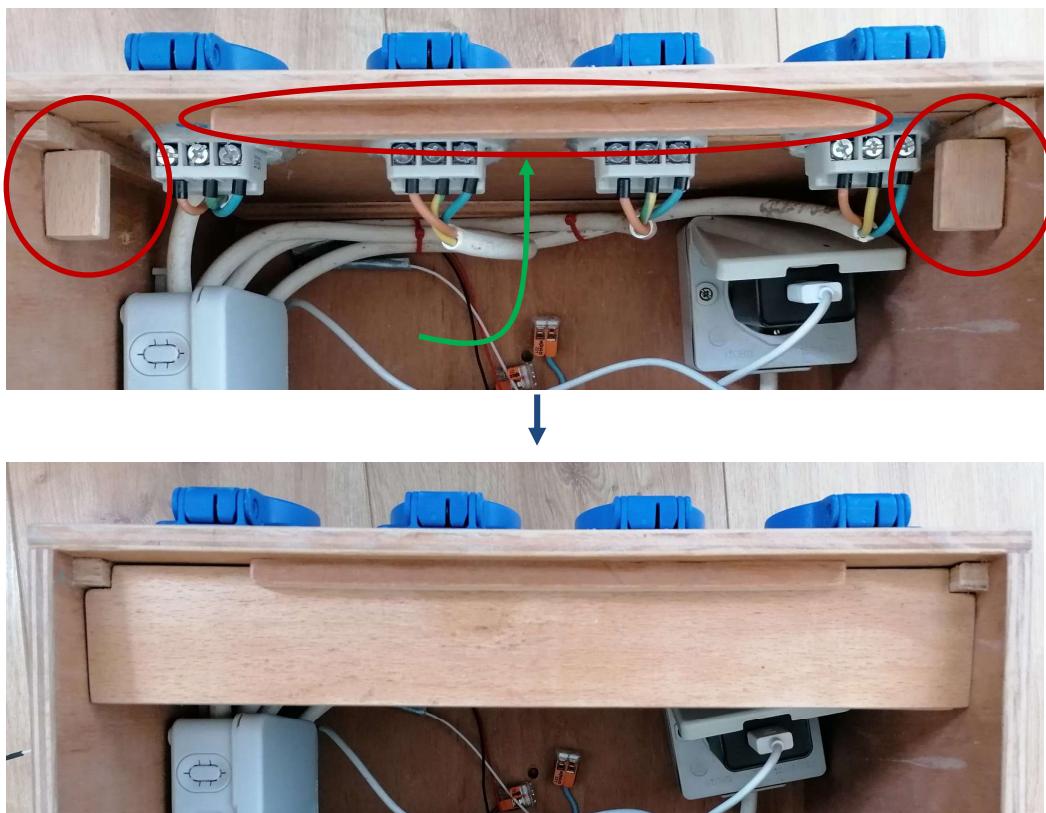


Abbildung 58: Schutzabdeckung der Steckdosenkontakte

Es ist jedoch zwingend darauf hinzuweisen, dass auch mit eingebrachter Schutzabdeckung kein vollständiger Schutz gewährleistet ist. Deshalb gilt bei Arbeiten innerhalb der Verteilerbox die Stromversorgung der Verteilerbox im Voraus auszustecken.

11.5.4 Relaisboard und Raspberry Pi

Das Relaisboard ist an der Rückwand der Verteilerbox mit einer Schraubenverbindung fest fixiert. Das verwendete Relaisboard (Waveshare Raspberry Pi 8-ch Relays → Link in Anhang 13.1) ermöglicht es, dass Raspberry Pi direkt über eine Kontaktleiste aufzustecken. Beide Komponenten sind in Abbildung 59 zusammen dargestellt. Um zusätzlich den DHT22 mit den entsprechenden GPIO Pins verbinden zu können, ist eine weitere Kontaktleiste auf dem Relaisboard angelötet.



Abbildung 59: Relaisboard und Raspberry Pi

Die gesamte elektrische Verkabelung des Steuerteils ist in Abbildung 60 dargestellt. In diesem Fall wurden auch die Verbindungen zwischen Relaisboard und Raspberry Pi durch Kabel dargestellt (nicht wie oben beschrieben über Kontaktleiste), um das Zusammenspiel zu verdeutlichen.

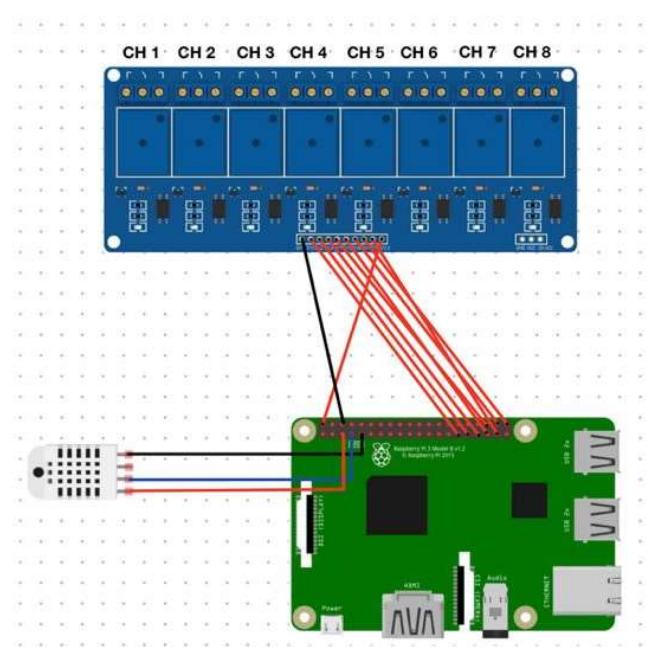


Abbildung 60: Verkabelung Steuerteil

11.6 Zusammenbau

In den folgenden Abbildungen ist der Zusammenbau des Fungariums abgebildet.

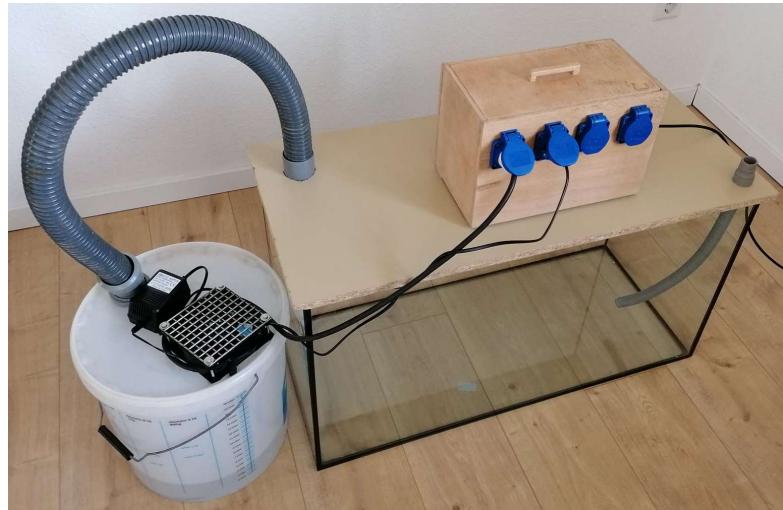


Abbildung 61: Zusammenbau Fungarium 1

Darauf ist die Verbindung vom Gesamtsystem (Aquarium) mit dem Lüftungssystem zu erkennen, die durch einen flexiblen Schlauch realisiert wird. Der Schlauch ist jeweils durch Klemmung mit den Rohrvorrichtungen verbunden. Die Kabel des Foggers und Lüfters sind in die Steckdosen der Verteilerbox eingesteckt und werden bei jeweiligem Betrieb vom Raspberry Pi über die Relais an -oder ausgeschalten.



Abbildung 62: Zusammenbau Fungarium 2

11.7 Weitere Komponenten

Die in dieser Konfiguration verwendeten Komponenten sowie deren Zusammenbau wurden in den letzten Abschnitten detailliert beschrieben. Dabei sind folgende Aktoren zur Manipulation der Umgebung der Pilz Brut in Verwendung:

- Lüftung → Luftzirkulation
- Fogger → Manipulation der Luftfeuchtigkeit
- Licht → Lichtquelle

Des Weiteren befindet sich ein DHT22 Sensor zur Messung der Temperatur und der Luftfeuchtigkeit im Einsatz.

Neben den bereits verwendeten Komponenten sind Erweiterungen zur Verbesserung der Automatisierung denkbar. Eine Erweiterung wäre beispielsweise der Einsatz einer Heizung, sowie einer Kühlung zur Steuerung der Temperatur. Diese optionale Erweiterung wurde im Projekt bereits sowohl von Seiten der Software als auch von Seiten der Hardware berücksichtigt. Sollte der Ausbau um einer der beiden Elemente vorgesehen werden, so kann das entsprechende Element über die Steuerseite angewählt werden. Anschließend ist lediglich ein Neustart des Raspberry Pi notwendig. Hardwarespezifisch wurde die optionale Erweiterung durch das Hinzufügen von zwei weiteren Steckdosen an der Verteilerbox berücksichtigt. Jedoch ist nicht nur eine Erweiterung um weitere Aktoren möglich, sondern auch die zusätzliche Ausstattung mit weiteren Sensoren hätten eine positive Auswirkung auf die Genauigkeit der Automatisierung und würden somit die Ausbeute der Pilzzucht steigern. Eine Möglichkeit wäre beispielsweise die Implementierung eines Bodenfeuchtesensors. Dadurch würde sich ein zusätzlicher Anhaltspunkt zur bereits gemessenen relativen Luftfeuchtigkeit ergeben. Zusätzlich wäre der Einsatz eines CO₂-Sensors zur Messung des CO₂-Gehalts von großem Vorteil, da speziell Pilze sehr anfällig für CO₂ sind. Für den Einsatz der Sensoren sind zusätzliche Änderungen in der Hardware sowie in der Software notwendig.

12 Fazit und Ausblick

Die Dokumentation beschreibt die vollständige Hardware- und Softwareentwicklung eines Hobby-Fungariums, welches gleichermaßen auch als Terrarium eingesetzt werden kann. Der gesamte Softwarecode steht auf GitHub unter folgender Adresse zu Verfügung: https://github.com/svenSchelling/Fungarium_Code

Für die hier beschriebene Gesamtapplikation, die voll einsatzfähig ist, wurde eine Vielzahl von Technologien verwendet. Aufgrund dessen hatten wir die Chance und Herausforderung uns in teilweise neuen und verschiedenen Fachgebieten einzuarbeiten. Gleichzeitig war der zeitliche Aufwand des Projekts vergleichsweise hoch. Im Falle einer reduzierten Anwendung (z.B. ohne Webserver mit externem Zugriff), wäre eine Vereinfachung des Systems möglich gewesen.

Auch über die Anforderungen des Auftraggebers hinaus wurden weitere hilfreiche Funktionen implementiert. Beispielsweise wurde zur automatisierten Installation aller Softwarefunktionen eine BASH-Datei erstellt. Diese vereinfacht und verkürzt vor allem den Installationsvorgang und die Konfiguration der Komponenten des Projekts.

Des Weiteren wurde zum Testen der einzelnen Komponenten ein manueller Modus integriert. Eine zusätzliche Funktion ist, das Einstellen der Systemkonfiguration (Bsp. GPIO Pins) vor dem Start der Applikation, sowie die Zustandsdiagramme der einzelnen Aktoren auf der Protokollseite.

Limitierungen der Applikation liegen im Bereich der Hardwareabhängigkeit. Die in dieser Dokumentation beschriebene Implementierung ist unter Verwendung eines Raspberry Pi möglich. Zur Umsetzung mithilfe eines anderen Mikrocontrollers sind Hardware -und Softwareanpassungen notwendig. Eine Hardwareunabhängigkeit war nicht vom Auftraggeber gefordert und war nicht das Ziel der Studienarbeit.

Mögliche Erweiterungen:

- Bodenfeuchtesensor zur Messung der Bodenfeuchte (als Erweiterung zur relativen Luftfeuchte)
- Heizung und Kühlung
- CO₂-Sensor

IV. Literaturverzeichnis

- Ellingwood, J. (10. August 2015). *digitalocean.com*. Von
<https://www.digitalocean.com/community/tutorials/how-to-set-up-password-authentication-with-apache-on-ubuntu-14-04> abgerufen
- Luber, D.-I. (13. März 2019). *cloudcomputing-insider*. Abgerufen am 22. Oktober 2020 von <https://www.cloudcomputing-insider.de/was bedeutet-lamp-a-808284/>
- o.V. (7. Juni 2018). *fuseon-media.com*. Abgerufen am 22. November 2020 von
<https://fuseon-media.com/ssl-verschluesselung-bedeutung-des-ssl-zertifikats-fuer-seo/>
- o.V. (2019). *ionos*. Von <https://www.ionos.de/digitalguide/domains/domaintipps/was-ist-eine-domain/> abgerufen
- o.V. (4. Juli 2019). *pimylifeup.com*. Von <https://pimylifeup.com/raspberry-pi-apache/> abgerufen
- o.V. (14. Juli 2019). *pimylifeup.com*. Von <https://pimylifeup.com/raspberry-pi-phpmyadmin/> abgerufen
- o.V. (10. März 2020). *ionos.de*. Abgerufen am 22. November 2020 von
<https://www.ionos.de/digitalguide/websites/webseiten-erstellen/wie-stelle-ich-meine-seite-auf-ssl-und-https-um/>
- o.V. (o. J.). *biteno.com*. Abgerufen am 22. Oktober 2020 von [biteno.com:](https://www.biteno.com/was-ist-apache/)
<https://www.biteno.com/was-ist-apache/>
- o.V. (o.J.). *certbot.eff*. Abgerufen am 22. November 2020 von
<https://certbot.eff.org/lets-encrypt/pip-apache>
- o.V. (o.J.). *certbot.eff.org*. Von <https://certbot.eff.org/lets-encrypt/pip-apache> abgerufen
- o.V. (o.J.). *datenbanken-verstehen.de*. Von <https://www.datenbanken-verstehen.de/lexikon/datenbanksystem/> abgerufen
- o.V. (o.J.). *ionos*. Von <https://www.ionos.de/digitalguide/server/knowhow/was-ist-ddns-dynamisches-dns/> abgerufen
- o.V. (o.J.). *php.net*. Abgerufen am 24. Oktober 2020 von
<https://www.php.net/manual/de/faq.general.php#faq.general.acronym>
- o.V. (o.J.). *raspberrypi.org*. Von <https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md> abgerufen

o.V. (o.J.). *techopedia*. Von <https://www.techopedia.com/definition/4057/port-forwarding> abgerufen

Oracle. (2020). *Class ArrayBlockingQueue*. Abgerufen am 10. 12 2020 von <https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/ArrayBlockingQueue.html>

Piazz, C. (15. Juni 2017). *Raspberry Pi - Temperatur und Luftfeuchtigkeit*. Abgerufen am 29. 12 2020 von <https://www.modius-techblog.de/mikrokontroller/raspberry-pi-temperatur-und-luftfeuchtigkeit/>

Stefan Luber, N. L. (6. Juni 2017). *bigdata-insider*. Abgerufen am 8. November 2020 von <https://www.bigdata-insider.de/was-ist-mysql-a-614184/>

The Apache Software Foundation. (06. 11 2020). *Logging Apache*. Abgerufen am 03. 01 2021 von <https://logging.apache.org/log4j/2.x/manual/configuration.html>

tutorialspoint. (2020). *Design Pattern - Singleton Pattern*. Abgerufen am 20. 12 2020 von https://www.tutorialspoint.com/design_pattern/singleton_pattern.htm

13 Anhang

13.1 Dokumentation der Steuerseite Use Cases

Name (ID)		Automatische Einstellungen übertragen (UC1)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Menüseite.
	V2	Die Voreinstellungen sind konfiguriert (UC6, UC7, UC8, UC9, UC10)
	V3	Bei der Auswahl des Steuerungsmodi (UC4) ist „Automatisch“ ausgewählt.
Normaler Ablauf	N1	<p>Der Benutzer gibt folgende Daten in die Eingabefelder (Blöcke) ein:</p> <ul style="list-style-type: none"> • <u>Licht Einstellungen</u>: Lichtstart, Lichtende • <u>Temperatur Einstellungen</u>: Mindesttemperatur, Höchsttemperatur, Mindesttemperatur Nacht, Höchsttemperatur Nacht • <u>Luftfeuchtigkeit Einstellungen</u>: Mindestfeuchtigkeit, Befeuchtungsdauer • <u>Lüftung Einstellungen</u>: Lüftungsdauer, Lüftungsintervall
	N2	Der Benutzer bestätigt die eingegebenen Daten mit einer der in den Einstellungsblöcken enthaltenen „Submit“-Buttons.
	N3	Ein Pop-Up-Fenster mit der Zusammenfassung der zu übertragenen Daten wird angezeigt.
	N4	Der Benutzer betätigt den „Ok“-Button des Pop-Up-Fensters und die Daten werden an die Datenbank übermittelt. Das Fenster schließt sich.
Nachbedingung		Die Webseite muss aktualisiert werden (wird automatisch nach der Übertragung durchgeführt). Damit werden neue Werte aus der Datenbank in die Eingabefelder geladen.
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht übermittelt werden und das System bleibt im vorherigen Zustand und benutzt die vorliegenden Einstellungsparameter.

Tabelle 4: Automatische Einstellungen übertragen

Name (ID)		Manuelles Starten vorhandener Aktoren (UC2)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Menüseite.
	V2	Die Voreinstellungen sind konfiguriert (UC6, UC7, UC8, UC9, UC10)
	V3	Bei der Auswahl des Steuerungsmodi (UC4) ist „Manuell“ ausgewählt.
	V4	Vorhandene Komponenten sind ausgewählt (UC8)
Normaler Ablauf	N1	Die vorhandenen Komponenten werden im Block „Manueller Aktorbetrieb“ angezeigt und können vom Benutzer eingeschaltet werden.
	N2	Der Benutzer betätigt folgende Checkboxen, wenn sie in UC8 ausgewählt sind: <ul style="list-style-type: none"> • Licht • Heizung • Kühlung • Fogger • Lüftung • Lüftung leicht • Sensor
	N3	Wird die Komponente eingeschalten, wird automatisch eine „1“ (An) an die Datenbank übertragen. Für das Ausschalten folgt eine „0“ (Aus).
Nachbedingung		-
Alternativer Ablauf	A1	Soll eine manuelle Steuerung einer Komponente im automatischen Modus durchgeführt werden, kann dies durch die Anpassung der Einstellungen geschehen. Wählt man beispielsweise eine niedrige Mindestfeuchtigkeit wird die Befeuchtungsroutine und somit der Lüfter und der Fogger gestartet.
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht übermittelt werden und das System bleibt im vorherigen Zustand.

Tabelle 5: Manuelles Starten vorhandener Aktoren

Name (ID)		Starten manueller Aktionen (UC3)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Menüseite.
	V2	Bei der Auswahl des Steuerungsmodi (UC4) ist „Manuell“ ausgewählt.
Normaler Ablauf	N1	Der Benutzer betätigt den Buttons für einen Sensor Neustart im Block „Manuelle Aktionen“.
	N2	Ein Pop-Up-Fenster mit der Information der Datenübertragung an die Datenbank wird angezeigt.
	N3	Der Benutzer betätigt den „Ok“-Button des Pop-Up-Fensters und eine „1“ wird an die Datenbank übermittelt, um dem Steuerungsprogramm mitzuteilen, dass die jeweilige Aktion gestartet wurde. Das Fenster schließt sich.
Alternativer Ablauf	A1	Der Benutzer betätigt den Button zum Start einer Befeuchtungsroutine im Block „Manuelle Aktionen“.
	A2	Ein Pop-Up-Fenster mit der Information der Datenübertragung an die Datenbank wird angezeigt.
	A3	Der Benutzer betätigt den „Ok“-Button des Pop-Up-Fensters und eine „1“ wird an die Datenbank übermittelt, um dem Steuerungsprogramm mitzuteilen, dass die jeweilige Aktion gestartet wurde. Das Fenster schließt sich.
Nachbedingung		-
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht übermittelt werden und das System bleibt im vorherigen Zustand.

Tabelle 6: Starten manueller Aktionen

Name (ID)		Auswahl des Steuerungsmodi (UC4)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Menüseite.
Normaler Ablauf	N1	Der Benutzer betätigt den Toggle-Switch mit dem Label „Automatisch/Manuell“ im Block „Voreinstellungen nach links.“
	N2	Das System wechselt in den automatischen Modus und es wird automatisch eine „0“ (Aus) an die Datenbank übertragen.
	N3	Die manuellen Einstellungen (UC2, UC3) verschwinden, um den automatischen Modus gegenüber dem manuellen Modus zu verriegeln.
	N4	Die automatischen Komponenteneinstellungen (UC1) werden angezeigt und können bearbeitet werden.
Nachbedingung		-
Alternativer Ablauf	A1	Der Benutzer betätigt den Toggle-Switch mit dem Label „Automatisch/Manuell“ im Block „Voreinstellungen nach rechts.“
	A2	Das System wechselt in den manuellen Modus und es wird automatisch eine „1“ (An) an die Datenbank übertragen.
	A3	Die automatischen Einstellungen (UC1) verschwinden, um den automatischen Modus gegenüber dem manuellen Modus zu verriegeln.
	A4	Die manuellen Einstellungen (UC2, UC3) werden angezeigt und können bearbeitet werden.
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht übermittelt werden und das System bleibt im vorherigen Zustand.

Tabelle 7: Auswahl des Steuermodi

Name (ID)		Voreinstellungen anzeigen (UC5)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Menüseite.
Normaler Ablauf	N1	Der Benutzer betätigt den Toggle-Switch mit dem Label „Voreinstellungen“ im Block „Voreinstellungen nach links.“
	N2	Auf der Menüseite verschwinden die Voreinstellungen (UC6, UC7, UC8, UC9, UC10).
Nachbedingung		-
Alternativer Ablauf	A1	Der Benutzer betätigt den Toggle-Switch mit dem Label „Voreinstellungen“ im Block „Voreinstellungen“ nach rechts.
	A2	Auf der Menüseite werden die Voreinstellungen (UC6, UC7, UC8, UC9, UC10) angezeigt und können bearbeitet werden.

Tabelle 8: Voreinstellung anzeigen

Name (ID)		Auswahl des Verwendungszwecks (UC6)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Menüseite.
	V2	Bei der Auswahl der Voreinstellungen (UC5) ist der Toggle-Switch nach rechts betätigt.
Normaler Ablauf	N1	Der Benutzer betätigt den Toggle-Switch mit dem Label „Terrarium/Fungarium“ im Block „Voreinstellungen nach links.“
	N2	Das Icon der PHP-Seiten in der Headerleiste wird zu einem Terrarium Icon geändert.
	N3	Der Text der PHP-Seiten in der Headerleiste wird zu „Terrarium“ geändert.
	N4	Das Favicon der PHP-Seite wird zu einem Terrarium Icon geändert.
Nachbedingung		-
Alternativer Ablauf	A1	Der Benutzer betätigt den Toggle-Switch mit dem Label „Terrarium/Fungarium“ im Block „Voreinstellungen nach rechts.“
	A2	Das Icon der PHP-Seiten in der Headerleiste wird zu einem Pilze Icon geändert.
	A3	Der Text der PHP-Seiten in der Headerleiste wird zu „Fungarium“ geändert.
	A4	Das Favicon der PHP-Seite wird zu einem Pilze Icon geändert.
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht übermittelt werden und das System bleibt im vorherigen Zustand.

Tabelle 9: Auswahl des Verwendungszwecks

Name (ID)		Auswahl der Lichtfunktion (UC7)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Menüseite.
	V2	Bei der Auswahl der Voreinstellungen (UC5) ist der Toggle-Switch nach rechts betätigt.
Normaler Ablauf	N1	Der Benutzer betätigt den Toggle-Switch mit dem Label „Licht als Heizung“ im Block „Voreinstellungen“ nach links.
	N2	Im System wird das Licht nicht als Heizung verwendet und es wird automatisch eine „0“ (Aus) an die Datenbank übertragen.
Nachbedingung		-
Alternativer Ablauf	A1	Der Benutzer betätigt den Toggle-Switch mit dem Label „Licht als Heizung“ nach links.
	A2	Im System wird das Licht als Heizung verwendet und es wird automatisch eine „1“ (An) an die Datenbank übertragen.
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht übermittelt werden und das System bleibt im vorherigen Zustand.

Tabelle 10: Auswahl der Lichtfunktion

Name (ID)		Auswahl der vorhandenen Komponenten (UC8)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Menüseite.
	V2	Bei der Auswahl der Voreinstellungen (UC5) ist der Toggle-Switch im Block „Vorhandene Komponenten“ nach rechts betätigt.
Normaler Ablauf	N1	Die vorhandenen Komponenten werden im Block „Vorhandene Komponenten“ angezeigt und können vom Benutzer ein- und ausgeschaltet werden.
	N2	<p>Der Benutzer betätigt folgende Toggle-Switch, um sie auszuwählen:</p> <ul style="list-style-type: none"> • Licht • Heizung • Kühlung • Fogger • Lüftung • Lüftung leicht • Sensor
	N3	Wird die Komponente eingeschalten, wird automatisch eine „1“ (Ausgewählt) an die Datenbank übertragen. Für das Ausschalten folgt eine „0“ (Nicht Ausgewählt).
	N4	Ausgewählte Komponenten aus N2 werden in UC2 angezeigt. Nicht ausgewählte Komponenten verschwinden (UC2)
	N5	Auf der Protokollseite werden Zustandsdiagramme der ausgewählten Komponenten angezeigt (UC18). Bei nicht ausgewählten Komponenten verschwindet das Diagramm.
Nachbedingung		-
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht übermittelt werden und das System bleibt im vorherigen Zustand.

Tabelle 11: Auswahl der vorhandenen Komponenten

Name (ID)		Pinbelegung einsehen (UC9)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Menüseite.
	V2	Bei der Auswahl der Voreinstellungen (UC5) ist der Toggle-Switch nach rechts betätigt.
Normaler Ablauf	N1	Der Benutzer klickt auf den Pfeil oder den Text: „WPI-Pinbelegung“ im Block „GPIO-Pinbelegung“.
	N2	Ein Bild, das die Pinbelegung eines Raspberry Pi mit WPI zeigt, erscheint im Dropdown-Menü.
Nachbedingung		-

Tabelle 12: Pinbelegung einsehen

Name (ID)		Auswahl der Pinbelegung (UC10)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Menüseite.
	V2	Bei der Auswahl der Voreinstellungen (UC5) ist der Toggle-Switch nach rechts betätigt.
	V3	Die Komponenten sind an den jeweiligen Pins angeschlossen.
Normaler Ablauf	N1	<p>Der Benutzer gibt folgende Pinbelegungen ein, falls die jeweilige Komponente vorhanden ist:</p> <ul style="list-style-type: none"> • Licht • Heizung • Kühlung • Fogger • Lüftung • Lüftung leicht • Sensor
	N2	Der Benutzer bestätigt die eingegebenen Daten mit dem „Submit“-Button.
	N3	Ein Pop-Up-Fenster mit der Zusammenfassung der zu übertragenen Pinbelegung wird angezeigt.
	N4	Der Benutzer betätigt den „Ok“-Button des Pop-Up-Fensters und die Daten werden an die Datenbank übermittelt. Das Fenster schließt sich.
Nachbedingung		Die Webseite muss aktualisiert werden (wird automatisch nach der Übertragung durchgeführt). Damit werden neue Werte aus der Datenbank in die Eingabefelder geladen.
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht übermittelt werden und das System bleibt im vorherigen Zustand und benutzt die vorliegende Pinbelegung.

Tabelle 13: Auswahl der Pinbelegung

Name (ID)		Temperaturverlauf in Diagramm anzeigen (UC11)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Temperaturseite.
Normaler Ablauf	N1	Die Temperaturdaten werden von der Datenbank gelesen und in eine JSON-Datei geschrieben.
	N2	Die JSON-Datei wird analysiert.
		Mit Hilfe des Moduls Chart.js wird ein Liniendiagramm mit der y-Achse Temperatur in °C und der x-Achse Uhrzeit erstellt.
	N3	Der Temperaturverlauf der letzten 24 Stunden wird in einem Liniendiagramm mit blauer Linie angezeigt.
	N4	Die eingestellte Mindesttemperatur und Höchsttemperatur werden in einem Liniendiagramm mit roter, horizontaler Linie angezeigt.
	N5	Die eingestellte Mindesttemperatur nachts und Höchsttemperatur nachts werden in einem Liniendiagramm mit orangener, horizontaler Linie angezeigt.
	N6	Der Benutzer kann die Linie im Diagramm vergrößern (zoomen) und verschieben.
	N7	Der Benutzer kann sich den Temperaturwert eines bestimmten Zeitpunktes über die Auswahl des Messpunktes im Diagramm in einem separaten Pop-Up-Fenster anzeigen lassen.
Nachbedingung		-
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht gelesen werden und das Diagramm bleibt leer.

Tabelle 14: Temperaturverlauf in Diagramm anzeigen

Name (ID)		Temperaturverlauf in Tabelle anzeigen (UC12)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Temperaturseite.
Normaler Ablauf	N1	Die Temperaturdaten werden von der Datenbank gelesen und in eine JSON-Datei geschrieben.
	N2	Die JSON-Datei wird analysiert.
		Mit Hilfe des Moduls Datatables.js wird eine Tabelle erstellt.
	N3	In die Tabelle werden zwei Spalten mit der Zeit im Format „dd.mm.yy hh:mm“ und der Temperatur in °C eingefügt.
	N4	Der Benutzer kann die Reihenfolge der Zeit und Temperatur durch das Betätigen der Pfeile neben den Spaltenbeschreibungen umkehren.
	N5	Der Benutzer kann durch Betätigen der einzelnen Seite zeitlich frühere Einträge ansehen.
Nachbedingung		-
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht gelesen werden und die Tabelle bleibt leer.

Tabelle 15: Temperaturverlauf in Tabelle anzeigen

Name (ID)		Feuchtigkeitsverlauf in Diagramm anzeigen (UC13)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Feuchtigkeitsseite.
Normaler Ablauf	N1	Die Luftfeuchtigkeitsdaten werden von der Datenbank gelesen und in eine JSON-Datei geschrieben.
	N2	Die JSON-Datei wird analysiert.
		Mit Hilfe des Moduls Chart.js wird ein Liniendiagramm mit der y-Achse relative Luftfeuchtigkeit in % und der x-Achse Uhrzeit erstellt.
	N3	Der Feuchtigkeitsverlauf der letzten 24 Stunden wird in einem Liniendiagramm mit blauer Linie angezeigt.
	N4	Die eingestellte Mindestfeuchtigkeit wird in einem Liniendiagramm mit roter, horizontaler Linie angezeigt.
	N5	Der Benutzer kann die Linie im Diagramm vergrößern (zoomen) und verschieben.
	N6	Der Benutzer kann sich den Feuchtigkeitswert eines bestimmten Zeitpunktes über die Auswahl des Messpunktes im Diagramm in einem separaten Pop-Up-Fenster anzeigen lassen.
Nachbedingung		-
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht gelesen werden und das Diagramm bleibt leer.

Tabelle 16: Feuchtigkeitsverlauf in Diagramm anzeigen

Name (ID)		Feuchtigkeitsverlauf in Tabelle anzeigen (UC14)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Feuchtigkeitsseite.
Normaler Ablauf	N1	Die Luftfeuchtigkeitsdaten werden von der Datenbank gelesen und in eine JSON-Datei geschrieben.
	N2	Die JSON-Datei wird analysiert.
		Mit Hilfe des Moduls Datatables.js wird eine Tabelle erstellt.
	N3	In die Tabelle werden zwei Spalten mit der Zeit im Format „dd.mm.yy hh:mm“ und der Luftfeuchte in % eingefügt.
	N4	Der Benutzer kann die Reihenfolge der Zeit und Feuchtigkeit durch das Betätigen der Pfeile neben den Spaltenbeschreibungen umkehren.
	N5	Der Benutzer kann durch Betätigen der einzelnen Seite zeitlich frühere Einträge ansehen.
Nachbedingung		-
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht gelesen werden und die Tabelle bleibt leer.

Tabelle 17: Feuchtigkeitsverlauf in Tabelle anzeigen

Name (ID)		Protokoll in Tabelle anzeigen (UC15)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Protokollseite.
Normaler Ablauf	N1	Die Protokolldaten werden von der Datenbank gelesen und in eine JSON-Datei geschrieben.
	N2	Die JSON-Datei wird analysiert.
	N3	In die Tabelle werden zwei Spalten mit der Zeit im Format „dd.mm.yy hh:mm“ und dem Protokolleintrag eingefügt.
	N4	Der Benutzer kann die Reihenfolge der Zeit und Einträge durch das Betätigen der Pfeile neben den Spaltenbeschreibungen umkehren.
	N5	Der Benutzer kann durch Betätigen der einzelnen Seite zeitlich frühere Einträge ansehen.
Nachbedingung		-
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht gelesen werden und die Tabelle bleibt leer.

Tabelle 18: Protokoll in Tabelle anzeigen

Name (ID)		Nach Protokolldaten suchen (UC16)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Protokollseite.
	V2	Protokolldaten befinden sich in der Tabelle (UC15).
Normaler Ablauf	N1	Der Benutzer gibt in die Suchleiste einen Protokolleintrag ein.
	N2	Die gesuchten Zeitpunkte werden in der Tabelle zusammen mit den Einträgen angezeigt.
Alternativer Ablauf	A1	Der Benutzer gibt in die Suchleiste eine Zeit ein.
	A2	Die gesuchten Zeitpunkte werden in der Tabelle zusammen mit den Einträgen angezeigt.
Nachbedingung		-
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht gelesen werden und die Tabelle bleibt leer.

Tabelle 19: Nach Protokolldaten suchen

Name (ID)		Zustand der Komponenten anzeigen (UC17)
Vorbedingung	V1	Der Benutzer befindet sich auf der PHP-Protokollseite.
	V2	Die vorhandenen Komponenten müssen auf der Menüseite in den Voreinstellungen ausgewählt sein (UC8).
Normaler Ablauf	N1	Die Zustandsdaten werden von der Datenbank gelesen und in eine JSON-Datei geschrieben.
	N2	Die JSON-Datei wird analysiert.
		Mit Hilfe des Moduls Chart.js wird jeweils ein Liniendiagramm per ausgewählter Komponente (UC8) mit der y-Achse An/Aus (1/0) und der x-Achse Uhrzeit erstellt und in einem separaten Block dargestellt.
	N3	Der Zustandsverlauf der letzten 24 Stunden wird in einem Liniendiagramm angezeigt.
	N4	Der Benutzer kann die Linie im Diagramm vergrößern (zoomen) und verschieben.
	N5	Der Benutzer kann sich den Zustand eines bestimmten Zeitpunktes über die Auswahl des Messpunktes im Diagramm in einem separaten Pop-Up-Fenster anzeigen lassen.
Nachbedingung		-
Fehler	F1	Der Webserver hat keine Verbindung zur Datenbank. Die Daten können nicht gelesen werden und die Tabelle bleibt leer.

Tabelle 20: Zustand der Komponenten anzeigen

13.2 Dokumentation der Steuerprogramm Use Cases

Name (ID)		Verbindung zur Datenbank aufbauen (UC18)
Vorbedingung	V1	Apache Server und Datenbank System läuft
	V2	Entweder ausgeführtes Bash-Skript mit Angabe von Datenbankbezeichnung, Benutzername und Passwort oder manuelles Ändern der genannten Attribute in der Shell-Datei „start_fungarium.sh“.
	V3	Richtige Ablage des JAR-files „mysql-connector-java-8.0.22.jar“ → lib Ordner. Oder Ändern des Pfades in der Start-Shell.
Normaler Ablauf	N1	Java-Programm baut unter Berücksichtigung der „mysql-connector-java-8.0.22.jar“ Datei eine Verbindung zur Datenbank auf.
Nachbedingung		-
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten aus der Datenbank gelesen werden, sowie keine Daten in die Datenbank geschrieben werden. Der Fehler wird protokolliert und das Programm wird beendet.

Tabelle 21: Verbindung zur Datenbank aufbauen

Name (ID)		Voreinstellungen einlesen (UC19)
Vorbedingung	V1	Eine stabile Verbindung zur Datenbank (UC18)
	V2	Der Anwender hat die Voreinstellungen über die Steuerseite konfiguriert.
	V3	Die vom Anwender vorgegebenen Daten sind in die Datenbank übernommen.
Normaler Ablauf	N1	Im ersten Schritt wird ausgelesen welche Komponenten im System vorhanden sind. → Vorgabe durch den Anwender.
	N2	Im nächsten Schritt wird die GPIO-Pin Nummer ausgelesen.
	N3	Auslesen der Auswahl Licht als Heizung.
Nachbedingung		-
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten aus der Datenbank gelesen werden.
	F2	Es ist kein DHT22 vorhanden. Somit können keine Sensordaten ausgelesen werden und die Automatisierung ist nicht möglich. Das Programm bricht ab.

Tabelle 22: Voreinstellungen Einlesen

Name (ID)		Systemkonfiguration setzen (UC20)
Vorbedingung	V1	Voreinstellungen von der Datenbank müssen im Voraus eingelesen sei. (UC19)
Normaler Ablauf	N1	Nachdem die GPIO Pins eingelesen sind, werden von den entsprechenden Komponenten die GPIO-Pins mit der zugehörigen Nummer nach der WPI-Pinbelegung als Output gesetzt.
	N2	Zusätzlich wird die „Shutdown Option“ so konfiguriert, dass bei Abbruch des Programms das Ausschalten der Aktoren sichergestellt ist.
	N3	Die Einstellung Licht als Heizung wird je nach Anwahl ins Java-Programm übernommen.
	N4	Sollte keine Lüftung niedertourig, eine Lüftung hochtourig wiederum verfügbar sein, so werden Lüftung niedertourig und hochtourig gleichgestellt.
Nachbedingung		-
Fehler	F1	Es treten Fehler beim Setzen der GPIO-Pins auf. Beispielsweise wenn ein Pin doppelt vergeben wird.

Tabelle 23: Systemkonfiguration setzen

Name (ID)		Datenbank „Protokoll“ aktualisieren (UC21)
Vorbedingung	V1	Eine stabile Verbindung zur Datenbank. (UC18)
Normaler Ablauf	N1	Mithilfe des konfigurierten Datenbankloggers (DBlog) wird eine Ausgabe innerhalb des Java-Programms getätigt.
	N2	Ein neuer Datensatz wird in der Protokoll Datenbank hinzugefügt. Die entsprechende Nachricht befindet sich in der Spalte „Eintrag“ der DB-Tabelle.
Nachbedingung		-
Alternativer Ablauf	A1	Einer der Akteure wird angesteuert. Mithilfe des konfigurierten Datenbankloggers (DBmap) wird eine Ausgabe innerhalb des Java-Programms getätigt.
	A2	Ein neuer Datensatz wird in der Protokoll Datenbank hinzugefügt. Dabei werden die Spalten „Bauteil“, „An“ und „Eintrag“ der Aktion entsprechend ausgefüllt.
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten in die Datenbank geschrieben werden.

Tabelle 24: Datenbank "Protokoll" aktualisieren

Name (ID)		Datenbank „Manuell“ aktualisieren (UC22)
Vorbedingung	V1	Eine stabile Verbindung zur Datenbank. (UC18)
Normaler Ablauf	N1	Ein Aktor wird zu einem Zustandswechsel gezwungen.
	N2	Nach dem Wechseln des Zustands wird der Datenbankeintrag des entsprechenden Aktors aktualisiert.
Nachbedingung		-
Alternativer Ablauf	A1	Die Befeuchtungsroutine oder ein Sensorneustart wird eingeleitet.
	A2	Nach dem Start einer der beiden Aktionen wird der Datenbankeintrag der entsprechenden Routine aktualisiert.
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten in die Datenbank geschrieben werden.

Tabelle 25: Datenbank "manuell" aktualisieren

Name (ID)		Sensordaten einlesen (UC23)
Vorbedingung	V1	Einlesen der Voreinstellungen (UC19)
	V2	Setzen Systemkonfiguration (UC20)
	V3	Richtige Ablage des Python-Skripts „mydht.py“ → lib Ordner. Oder Ändern des Pfades in der Start-Shell.
Normaler Ablauf	N1	Im Normalbetrieb wird alle 120 s eine Messung gestartet. Dabei wird im ersten Schritt das Python-Skript aufgerufen.
	N2	Python-Skript liest die Daten vom DHT22 und schreibt diese in die Systemkonsole.
	N3	Das Java-Programm liest die vom Python-Skript in der Systemkonsole ausgegeben Daten ein.
Nachbedingung		-
Alternativer Ablauf	A1	Zusammenspiel zwischen Python Skript und Java-Programm über die Systemkonsole ist fehlerbehaftet.
	A2	Bei Fehlern werden die vorherigen Sensordaten zur Weiterverarbeitung verwendet.
Fehler	F1	Aufruf des Python Skripts funktioniert nicht.
	F2	Sensor ist nicht richtig mit dem Raspberry Pi verbunden.
	F3	Kabellänge zwischen Raspberry Pi und Sensor ist zu groß gewählt. Sensordaten können nicht ausgelesen werden.
	F4	Sensor ist kaputt und gibt unplausible Daten aus.

Tabelle 26: Sensordaten einlesen

Name (ID)		Sensordaten weiterverarbeiten (UC24)
Vorbedingung	V1	Sensordaten müssen eingelesen sein (UC23)
	V2	Eine stabile Verbindung zur Datenbank. (UC19)
Normaler Ablauf	N1	Sensordaten werden auf deren Plausibilität geprüft.
	N2	Nach erfolgreicher Prüfung werden die aufgenommenen Sensordaten in die Datenbank geschrieben.
	N3	Sensordaten werden mit den aktuellen Einstellparametern aus der Datenbank verglichen. (Temperatur und Luftfeuchtigkeit)
	N4	Anschließend werden bei Bedarf die entsprechenden Manipulationskomponenten angesteuert.
Nachbedingung		-
Alternativer Ablauf	A1	Bei fehlerhafter Plausibilitätsprüfung wird ein Sensorneustart eingeleitet.
	A2	Nach drei fehlerhaften Messungen in Folge wird das System in den Notbetrieb versetzt. Dabei werden die Aktoren Heizung und Kühlung ausgeschalten. Des Weiteren wird eine regelmäßige Befeuchtungsroutine in Zeitabständen von 15 min eingeleitet.
Fehler	F1	Es werden keine neuen Sensordaten aufgenommen.

Tabelle 27: Sensordaten weiterverarbeiten

Name (ID)		Sensorneustart (UC25)
Vorbedingung	V1	Einlesen der Voreinstellungen (UC19)
	V2	Setzen Systemkonfiguration (UC20)
Normaler Ablauf	N1	Im Normalbetrieb wird der Sensorneustart eingeleitet, wenn die Plausibilitätsprüfung der Sensordaten erfolglos war. Der Sensor wird ebenfalls alle 24 h Stunden um „00:00 Uhr“ neugestartet. Des Weiteren ist es möglich im manuellen Betrieb ein Sensorneustart einzuleiten.
	N2	Der Sensor wird von der Spannungsversorgung getrennt.
	N3	Nach 30 s wird der Sensor wieder eingeschalten.
	N4	Nach einer weiteren Wartezeit von 30s wird eine Messung eingeleitet.
Nachbedingung		-
Alternativer Ablauf	A1	Im Notbetrieb startet der Sensor alle 60 min neu. Zusätzlich kann auch im Notbetrieb ein Sensorneustart manuell eingeleitet werden.
	A2	Der Sensor wird von der Spannungsversorgung getrennt.
	A3	Nach 5min wird der Sensor wieder eingeschalten.
	A4	Nach einer weiteren Wartezeit von 30 s wird eine Messung eingeleitet.
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten von der Datenbank gelesen werden und somit ist ein Einschalten im manuellen Betrieb nicht möglich.

Tabelle 28: Sensorneustart

Name (ID)		Befeuchtungsroutine starten (UC26)
Vorbedingung	V1	Einlesen der Voreinstellungen (UC19)
	V2	Setzen Systemkonfiguration (UC20)
	V3	Fogger und Lüfter müssen vorhanden sein.
Normaler Ablauf	N1	Im Normalbetrieb wird die Befeuchtungsroutine eingeleitet, wenn die aktuelle Luftfeuchtigkeit unter der Mindestfeuchtigkeit liegt und die Lüftungsroutine bereits durchgelaufen ist. Im Falle einer Dauerlüftung wird bei zwei aufeinanderfolgenden Messungen, bei denen die aktuelle Luftfeuchtigkeit unter der Mindestfeuchtigkeit liegt, die Befeuchtungsroutine gestartet. Des Weiteren ist es möglich im manuellen Betrieb die Befeuchtungsroutine einzuleiten.
	N2	Im ersten Schritt werden Fogger und Lüftung niedertourig eingeschalten.
	N3	Anschließend wird über die durch den Anwender vorgegebene Befeuchtungsdauer abgewartet, bis beide Komponenten wieder ausgeschalten werden.
Nachbedingung		Kein frühzeitiger Abbruch durch den Wechsel vom manuellen Betrieb in den automatischen Betrieb.
Alternativer Ablauf	A1	Im Notbetrieb wird eine regelmäßige Lüftung gefordert. Dabei wird die Befeuchtungsroutine alle 15 min gestartet.
	A2	Im ersten Schritt werden Fogger und Lüftung niedertourig eingeschalten.
	A3	Anschließend wird über die durch den Anwender vorgegebene Befeuchtungsdauer abgewartet, bis beide Komponenten wieder ausgeschalten werden.
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten von der Datenbank gelesen werden und somit ist ein Einschalten im manuellen Betrieb nicht möglich.

Tabelle 29: Befeuchtungsroutine starten

Name (ID)		Fogger steuern (UC27)
Vorbedingung	V1	Einlesen der Voreinstellungen (UC19)
	V2	Setzen Systemkonfiguration (UC20)
	V3	Fogger muss vorhanden sein.
Normaler Ablauf	N1	Der Fogger wird immer angesteuert, wenn es die Befeuchtungsroutine erfordert. Des Weiteren ist es möglich den Fogger im manuellen Betrieb ein- und auszuschalten.
Nachbedingung		-
Alternativer Ablauf	A1	Der Fogger wird beim Wechseln der manuellen Betriebsart in die automatische ausgeschalten.
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten von der Datenbank gelesen werden und somit ist ein Einschalten im manuellen Betrieb nicht möglich.

Tabelle 30: Fogger steuern

Name (ID)		Lüftung niedertourig steuern (UC28)
Vorbedingung	V1	Einlesen der Voreinstellungen (UC19)
	V2	Setzen Systemkonfiguration (UC20)
	V3	Lüftung niedertourig muss vorhanden sein.
Normaler Ablauf	N1	Die Lüftung niedertourig wird immer angesteuert, wenn es die Befeuchtungsroutine erfordert. Des Weiteren ist es möglich die niedertourige Lüftung im manuellen Betrieb ein- und auszuschalten.
Nachbedingung		Lüftung niedertourig und Lüftung hochtourig dürfen nicht gleichzeitig angesteuert werden. Die Befeuchtungsroutine hat eine höhere Priorität. Bedeutet sollte der Lüfter hochtourig laufen und die Befeuchtungsroutine erfordert ein Anschalten des niedertourigen Lüfters, so wird der hochtourige Lüfter im Voraus ausgeschalten.
Alternativer Ablauf	A1	Die Lüftung niedertourig wird beim Wechseln der manuellen Betriebsart in die automatische ausgeschalten.
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten von der Datenbank gelesen werden und somit ist ein Einschalten im manuellen Betrieb nicht möglich.

Tabelle 31: Lüftung niedertourig steuern

Name (ID)		Lüftungsroutine starten (UC29)
Vorbedingung	V1	Einlesen der Voreinstellungen (UC19)
	V2	Setzen Systemkonfiguration (UC20)
	V3	Lüfter muss vorhanden sein.
Normaler Ablauf	N1	Im Normalbetrieb wird die Lüftung eingeleitet, wenn in einer Messung die aktuelle Luftfeuchtigkeit unter der Mindestfeuchtigkeit liegt. Des Weiteren gibt es eine regelmäßige Lüftung. Das Lüftungsintervall wird durch den Anwender vorgegeben.
	N2	Im ersten Schritt wird der Lüfter hochtourig eingeschalten.
	N3	Anschließend wird über die durch den Anwender vorgegebene Lüftungsdauer abgewartet, bis der Lüfter niedertourig wieder ausgeschalten wird.
Nachbedingung		Kein frühzeitiges Ausschalten des Lüfters hochtourig, aufgrund des Einschalten des Lüfters niedertourig. Lüfter hochtourig und Lüfter niedertourig dürfen nicht gleichzeitig an sein
Fehler	F1	-

Tabelle 32: Lüftung starten

Name (ID)		Lüftung hochtourig steuern (UC30)
Vorbedingung	V1	Einlesen der Voreinstellungen (UC19)
	V2	Setzen Systemkonfiguration (UC20)
	V3	Lüftung hochtourig muss vorhanden sein.
Normaler Ablauf	N1	Die Lüftung hochtourig wird immer angesteuert, wenn es die Lüftungsroutine erfordert. Des Weiteren ist es möglich die hochtourige Lüftung im manuellen Betrieb ein- und auszuschalten.
Nachbedingung		Lüftung niedertourig und Lüftung hochtourig dürfen nicht gleichzeitig angesteuert werden. Die Befeuchtungsroutine hat eine höhere Priorität. Bedeutet sollte der Lüfter hochtourig laufen und die Befeuchtungsroutine erfordert ein Anschalten des niedertourigen Lüfters, so wird der hochtourige Lüfter im Voraus ausgeschalten.
Alternativer Ablauf	A1	Wenn keine niedertourige Lüftung vorhanden ist, so wird der hochtourige Lüfter anstelle des niedertourigen Lüfters angesteuert.
	A2	Wird der Lüfter auch Anstelle der Kühlung verwendet, so wird der Lüfter immer dann angesteuert, wenn die aktuelle Temperatur über die Maximaltemperatur ansteigt.
	A3	Der Lüfter wird wieder ausgeschalten, wenn die Temperatur wieder unter der gegebenen Schwelle liegt. (siehe Absatz 2.1.3)
	A4	Der Lüfter wird ausgeschalten beim Wechseln von manuellen Betriebsart in die automatische.
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten von der Datenbank gelesen werden und somit ist ein Einschalten im manuellen Betrieb nicht möglich.

Tabelle 33: Lüftung hochtourig steuern

Name (ID)		Licht steuern (UC31)
Vorbedingung	V1	Einlesen der Voreinstellungen (UC19)
	V2	Setzen Systemkonfiguration (UC20)
	V3	Lichtquelle muss vorhanden sein.
Normaler Ablauf	N1	Im Normalbetrieb wird die Lichtquelle immer angesteuert, wenn die aktuelle Uhrzeit zwischen dem durch den Anwender vorgegebenen Lichtstart und Lichtende liegt. Diese Bedingung wird alle 3s gecheckt. Des Weiteren ist es möglich im manuellen Betrieb das Licht einzuschalten. Im manuellen Betrieb wird das Licht nicht nach den Bedingungen durch die Uhrzeiten eingeschalten.
Nachbedingung		-
Alternativer Ablauf	A1	Wird das Licht auch an Stelle der Heizung verwendet, so wird das Licht immer dann angesteuert, wenn die aktuelle Temperatur unter die Mindesttemperatur fällt.
	A2	Das Licht wird wieder ausgeschalten, wenn die Temperatur wieder über der gegebenen Schwelle liegt. (siehe Absatz 2.1.3), sofern kein Einschalten durch den normalen Ablauf gegeben ist.
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten von der Datenbank gelesen werden und somit ist ein Einschalten im manuellen Betrieb nicht möglich.

Tabelle 34: *Licht steuern*

Name (ID)		Heizung steuern (UC32)
Vorbedingung	V1	Einlesen der Voreinstellungen (UC19)
	V2	Setzen Systemkonfiguration (UC20)
	V3	Heizung muss vorhanden sein.
Normaler Ablauf	N1	Die Heizung wird immer angesteuert, wenn die aktuelle Temperatur unter die Minimaltemperatur sinkt. Des Weiteren ist es möglich die Heizung im manuellen Betrieb ein- und auszuschalten.
	N2	Die Heizung wird ausgeschalten, wenn die Temperatur wieder über der gegebenen Schwelle liegt. (siehe Absatz 2.1.3)
Nachbedingung		-
Alternativer Ablauf	A1	Die Heizung wird ausgeschalten beim Wechsel in den Notbetrieb.
	A2	Die Heizung wird ausgeschalten beim Wechseln von manuellen Betriebsart in die automatische.
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten von der Datenbank gelesen werden und somit ist ein Einschalten im manuellen Betrieb nicht möglich.

Tabelle 35: Heizung steuern

Name (ID)		Kühlung steuern (UC33)
Vorbedingung	V1	Einlesen der Voreinstellungen (UC19)
	V2	Setzen Systemkonfiguration (UC20)
	V3	Kühlung muss vorhanden sein.
Normaler Ablauf	N1	Die Kühlung wird immer angesteuert, wenn die aktuelle Temperatur über die Maximaltemperatur ansteigt. Des Weiteren ist es möglich die Kühlung im manuellen Betrieb ein- und auszuschalten.
	N2	Die Kühlung wird ausgeschalten, wenn die Temperatur wieder unter der gegebenen Schwelle liegt. (siehe Absatz 2.1.3)
Nachbedingung		-
Alternativer Ablauf	A1	Die Kühlung wird ausgeschalten beim Wechsel in den Notbetrieb.
	A2	Die Kühlung wird ausgeschalten beim Wechseln von manuellen Betriebsart in die automatische.
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten von der Datenbank gelesen werden und somit ist ein Einschalten im manuellen Betrieb nicht möglich.

Tabelle 36: Kühlung steuern

Name (ID)		Protokollieren in Datei (UC34)
Vorbedingung	V1	-
Normaler Ablauf	N1	Mithilfe des konfigurierten Datenbankloggers (DBlog) wird eine Ausgabe innerhalb des Java-Programms getätigt.
	N2	Es wird eine neue Zeile in die log-Datei geschrieben.
	N3	Um 00:00 Uhr wird die log-Datei in einen Ordner namens „history“ als zip-Datei verschoben.
	N4	Neue log-Datei wird erstellt.
Nachbedingung		-
Alternativer Ablauf	A1	Mithilfe des konfigurierten Standartloggers des packages „com.dhbw.fungarium“ wird eine Ausgabe innerhalb des Java-Programms getätigt.
	A2	Es wird eine neue Zeile in die log-Datei geschrieben.
	A3	Um 00:00 Uhr wird die log-Datei in einen Ordner namens „history“ als zip-Datei verschoben.
	A4	Neue log-Datei wird erstellt.
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch ist der Datenbanklogger nicht verfügbar.

Tabelle 37: Protokollieren in Datei

Name (ID)		Wechsel der Betriebsart (UC35)
Vorbedingung	V1	Eine stabile Verbindung zur Datenbank (UC18)
	V2	Der Anwender hat ein Wechsel der Betriebsart angefordert.
	V3	Die vom Anwender vorgegebenen Daten sind in die Datenbank übernommen wurden.
Normaler Ablauf	N1	Die aktuelle Betriebsart wird alle 3 s vom Java-Programm ausgelesen.
Nachbedingung		-
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten aus der Datenbank gelesen werden.

Tabelle 38: Wechsel der Betriebsart

Name (ID)		Wechsel von Normalbetrieb in Notbetrieb (UC36)
Vorbedingung	V1	Drei fehlerhafte Plausibilitätsprüfungen der Sensordaten in Folge.
	V2	System im Normalbetrieb
Normaler Ablauf	N1	Nach drei fehlerhaften Messungen wird das System in Notbetrieb versetzt.
	N2	Heizung und Kühlung werden ausgeschalten.
Nachbedingung		-
Fehler	F1	-

Tabelle 39: Wechsel von Normalbetrieb in Notbetrieb

Name (ID)		Wechsel von Notbetrieb in Normalbetrieb (UC37)
Vorbedingung	V1	Erfolgreiche Plausibilitätsprüfung der Sensordaten.
	V2	System im Notbetrieb
Normaler Ablauf	N1	Befindet sich das System im Notbetrieb und es wird nach einem Sensorneustart eine erfolgreiche Messung aufgenommen, so wird vom Notbetrieb in den Normalbetrieb gewechselt.
Nachbedingung		-
Fehler	F1	-

Tabelle 40: Wechsel von Notbetrieb in Normalbetrieb

Name (ID)		Einstellparameter einlesen (UC38)
Vorbedingung	V1	Eine stabile Verbindung zur Datenbank (UC18)
Normaler Ablauf	N1	Die Einstellparameter werden in Zeitabständen 3 s eingelesen.
	N2	Befindet sich das System im Zustand Tag so werden die Temperatureinstellung für den Tag aus der Datenbank gelesen.
	N3	Befindet sich das System im Zustand Nacht so werden die Temperatureinstellungen für die Nacht aus der Datenbank gelesen.
Nachbedingung		-
Fehler	F1	Das Java Programm hat keine Verbindung zur Datenbank. Dadurch können keine Daten aus der Datenbank gelesen werden.

Tabelle 41: Einstellparameter einlesen

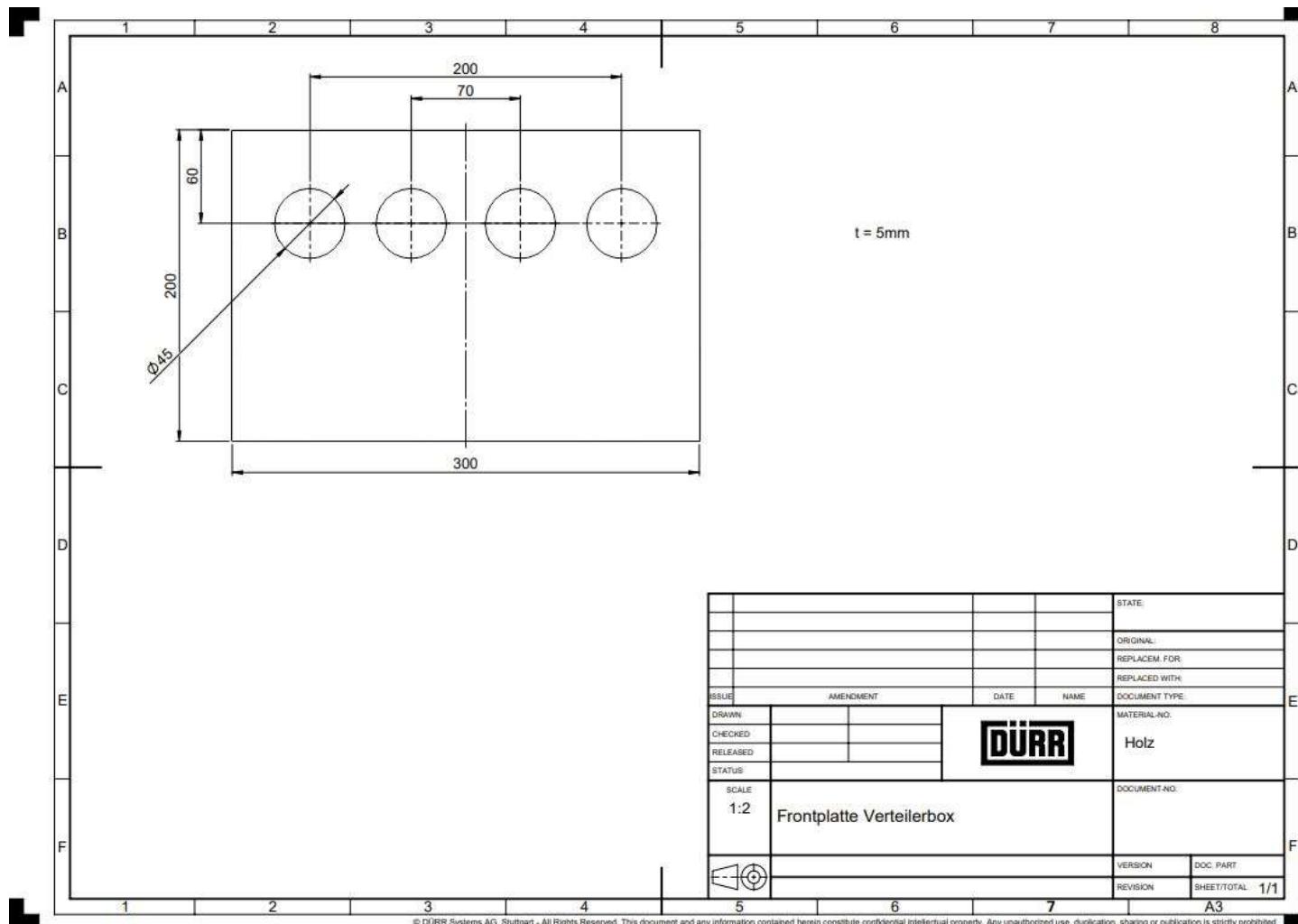
13.3 Komponentenliste

Komponente	Internetlink
Hygro-Plus Terrarien-Nebler	https://www.amazon.de/Hobby-37250-Hygro-Plus-Terrariennebler/dp/B000RYSOYI
DHT22	https://www.amazon.de/AZDelivery-Temperatursensor-Luftfeuchtigkeitssensor-Platine-Arduino/dp/B078SVZB1X/ref=sr_1_1_sspa?dchild=1&keywords=DHT22&qid=1609852810&sr=8-1-spons&psc=1&smid=A1X7QLRQH87QA3&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUFTRjVXTUJaT1BPSzMmZW5jcnlwdGVkSWQ9QTA5NTAyMzMyVVo0WUVMUkYxWkhYJmVuY3J5cHRIZEFkSWQ9QTA4NDY1NTUxRIdDUTVZSkhJWU8md2IkZ2V0TmFtZT1zcF9hdGYmYWN0aW9uPWNsaWNrUmVkaXJIY3QmZG9Ob3RMb2dDbGljaz10cnVI
Waveshare Raspberry Pi 8-ch Relays (Relaisboard)	https://www.amazon.de/dp/B07J3YNVG8/ref=cm_sw_r_c�_apai_uq2CFbJCAKBK4?th=1
Raspberry Pi 4 Modell B; 4 GB	https://www.amazon.de/Raspberry-Pi-ARM-Cortex-A72-Bluetooth-Micro-HDMI/dp/B07TC2BK1X/ref=sr_1_3?__mk_de_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&cid=3SWT3WM75FODE&dchild=1&keywords=raspberry+pi+4&qid=1609852906&quartzVehicle=812-409&replacementKeywords=raspberry+pi&sprefix=ras%2Caps%2C323&sr=8-3
Steckdosen	https://www.conrad.de/de/p/pce-601-450-06-anbau-steckdose-ip54-blau-624008.html
Lichtquelle	https://www.conrad.de/de/p/barthelme-gluehlampe-230-v-e27-15-w-gluehlampenform-inhalt-1-st-1313789.html

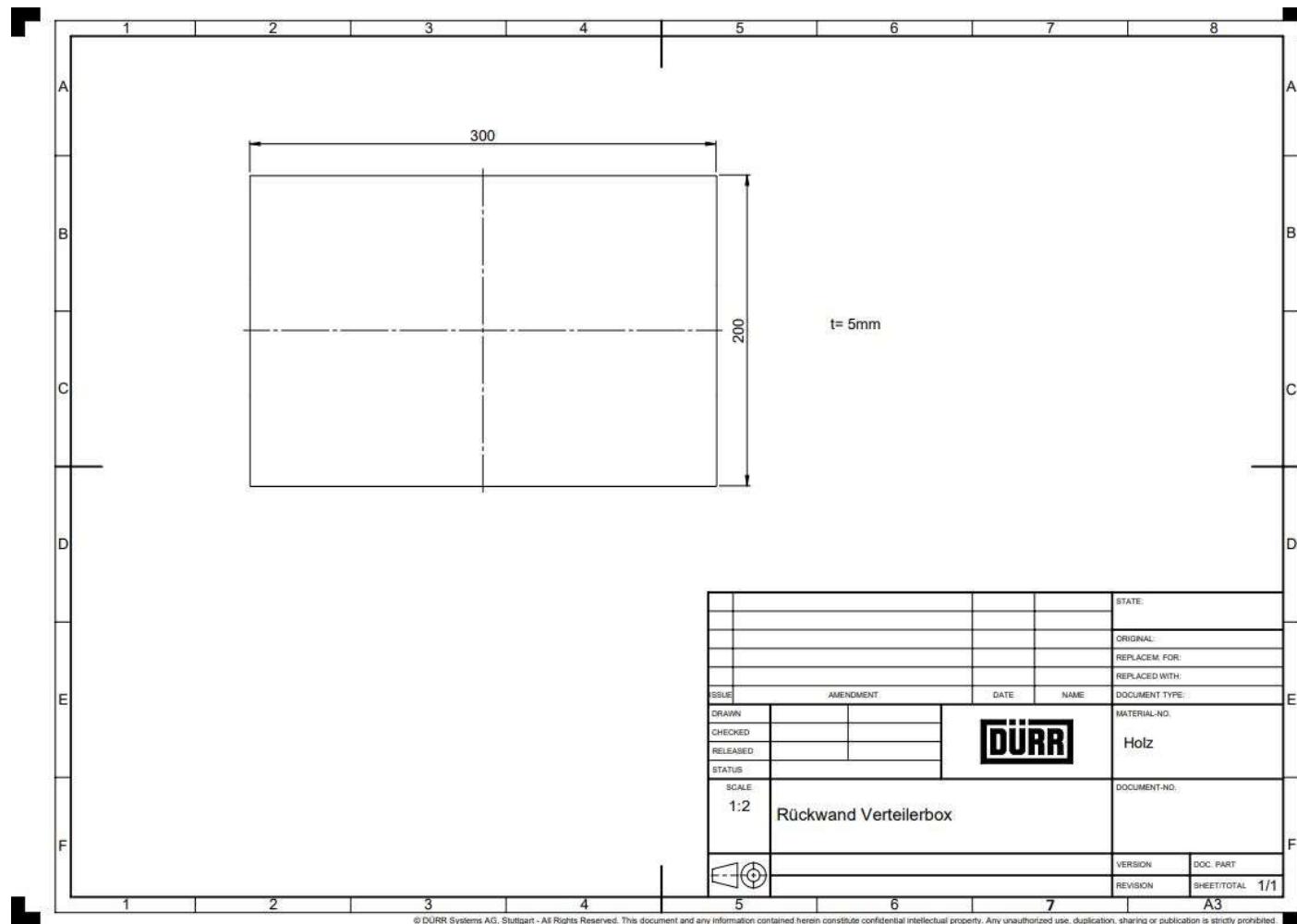
Tabelle 42: Komponentenliste Anhang

13.4 Fertigungszeichnungen

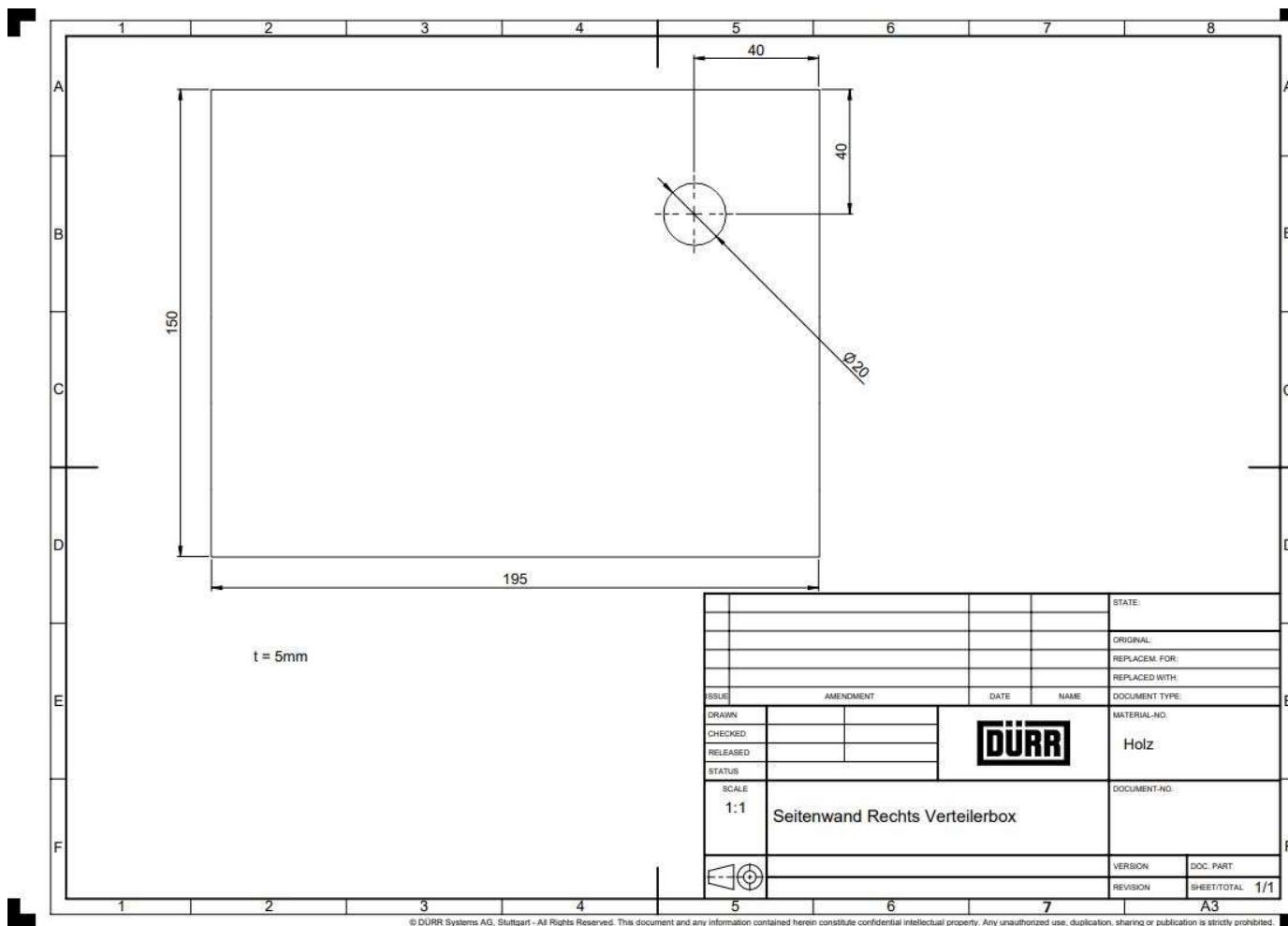
Frontplatte



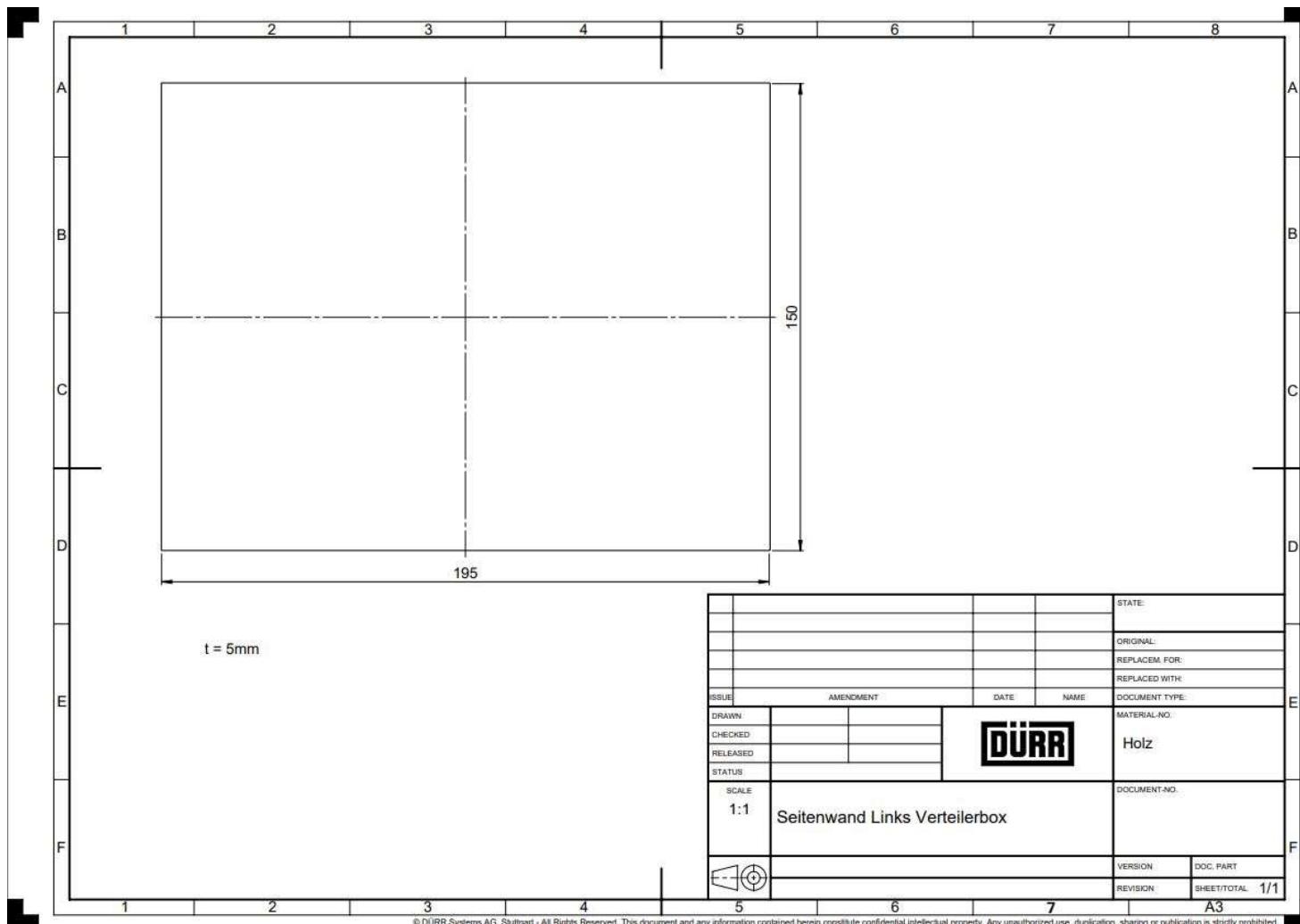
Rückwand



Seitenwand rechts

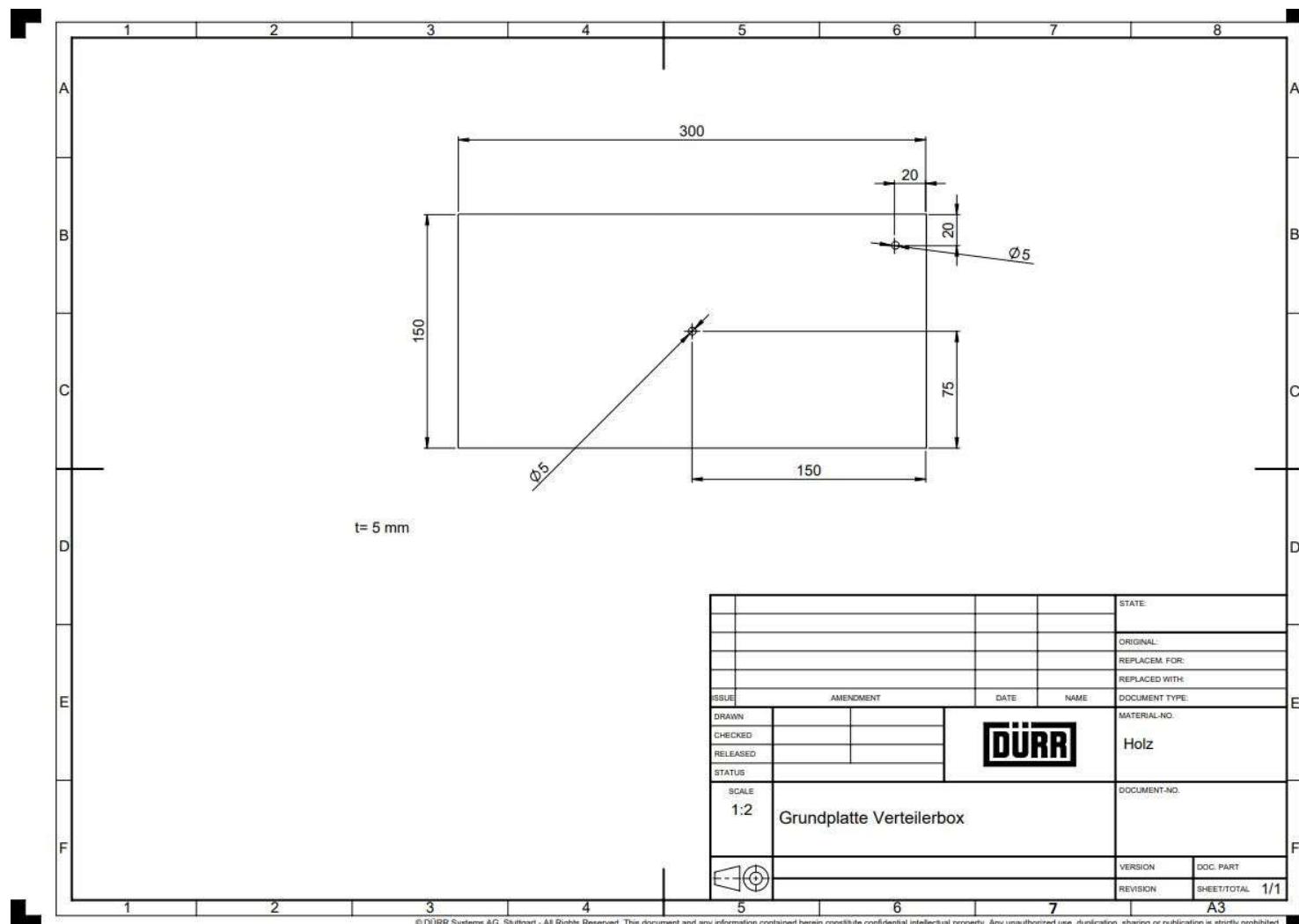


Seitenwand links

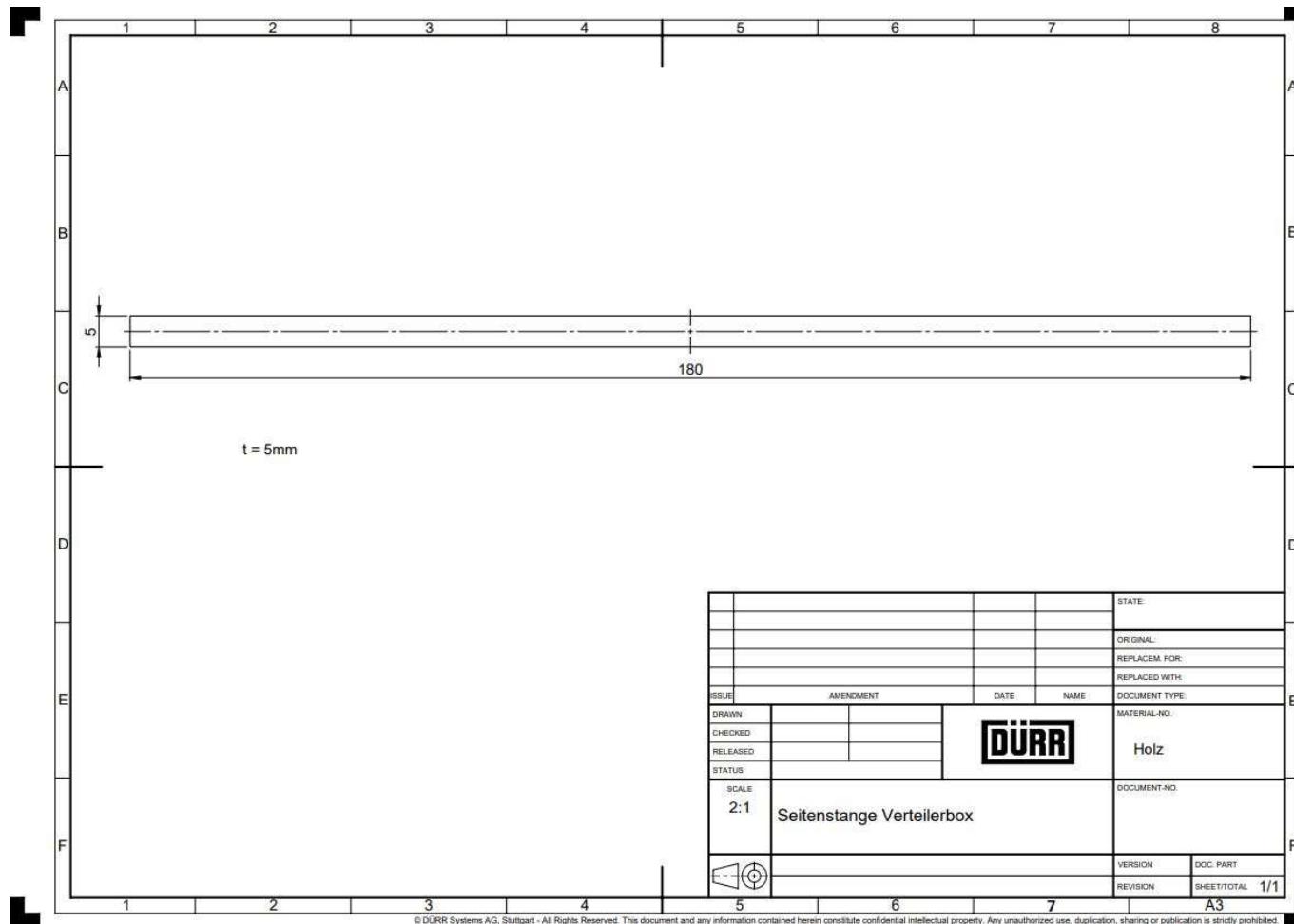


LV

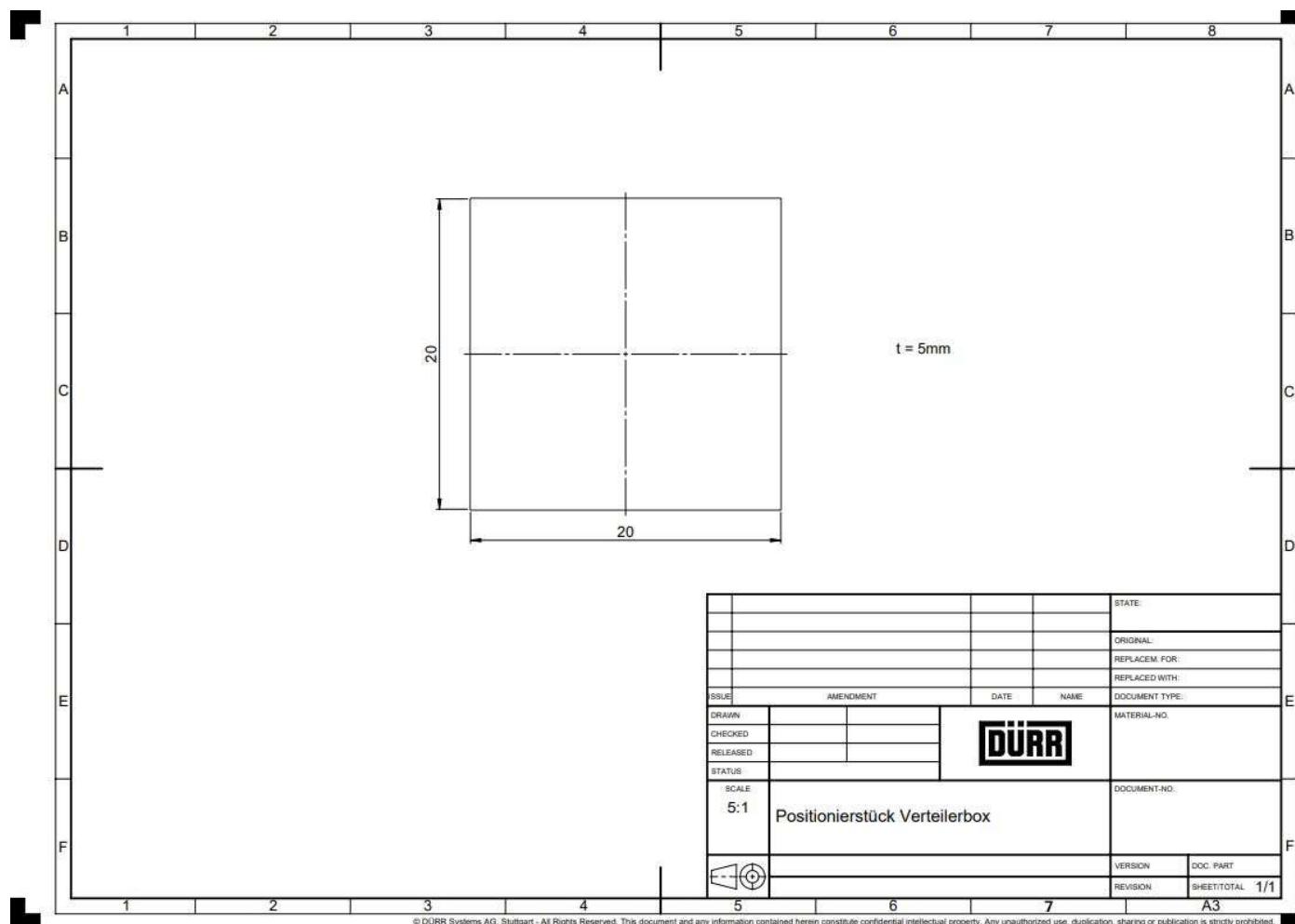
Grundplatte



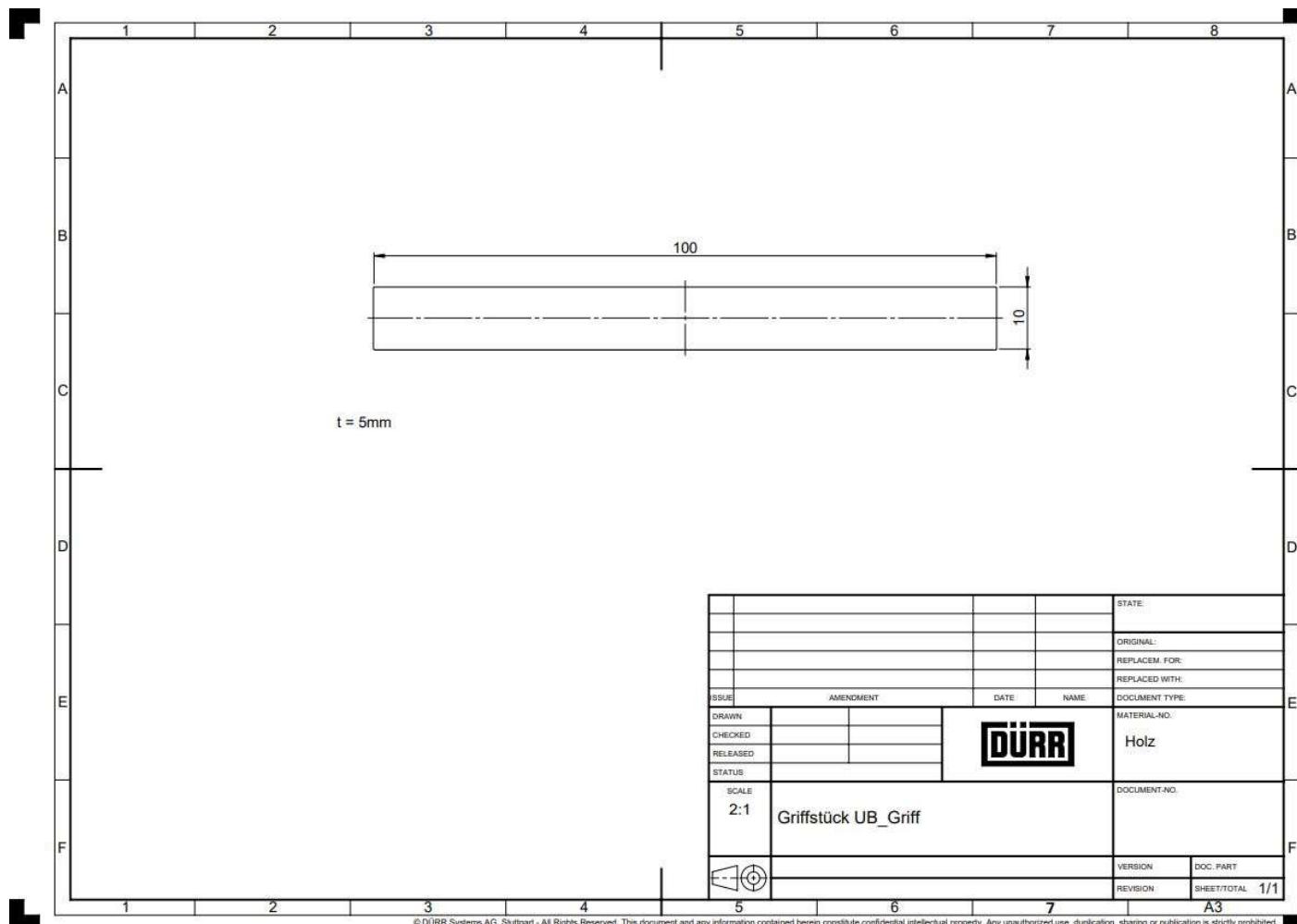
Seitenstangen → Stabilisation



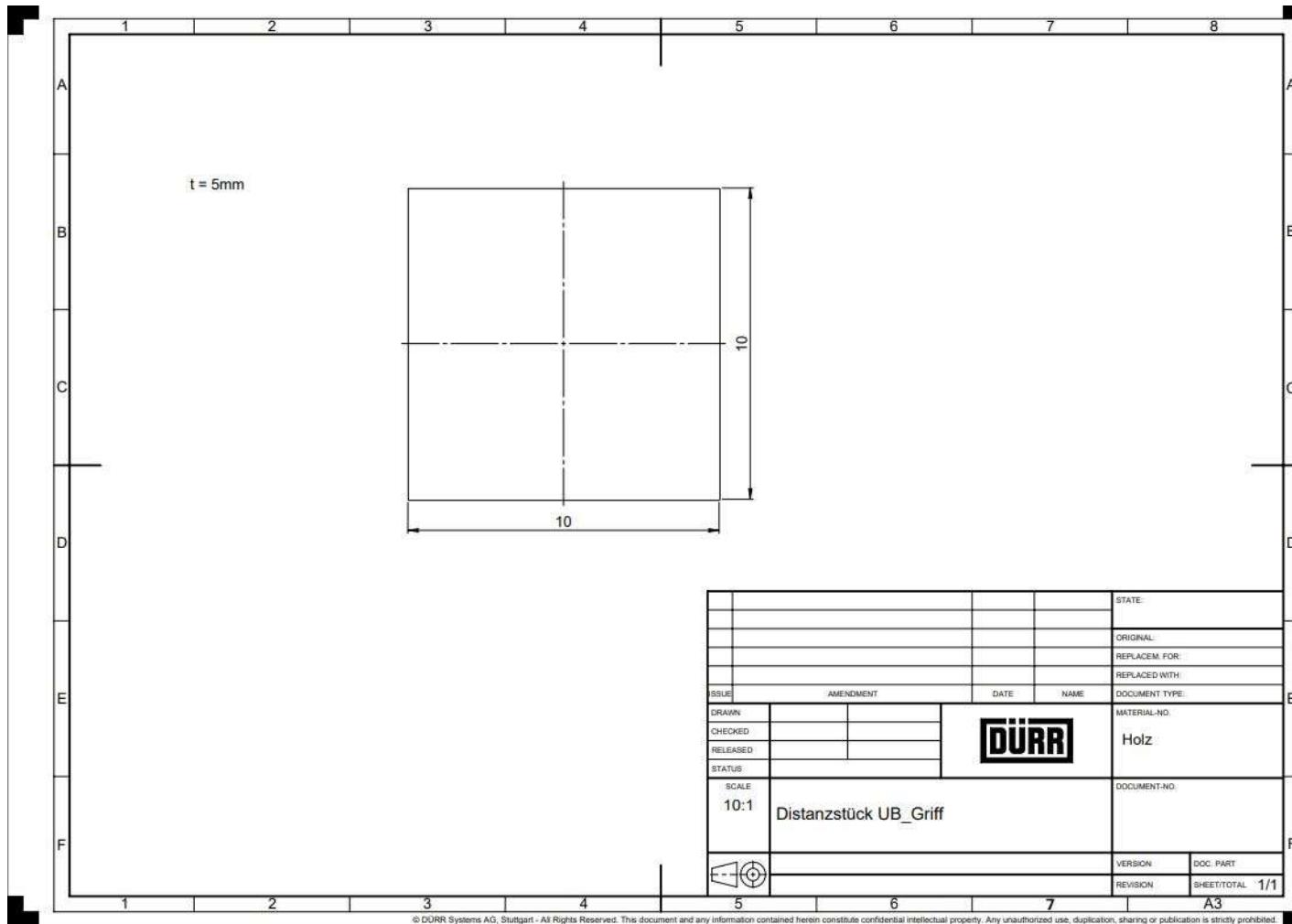
Positionierstück/Halterung Schutzabdeckung



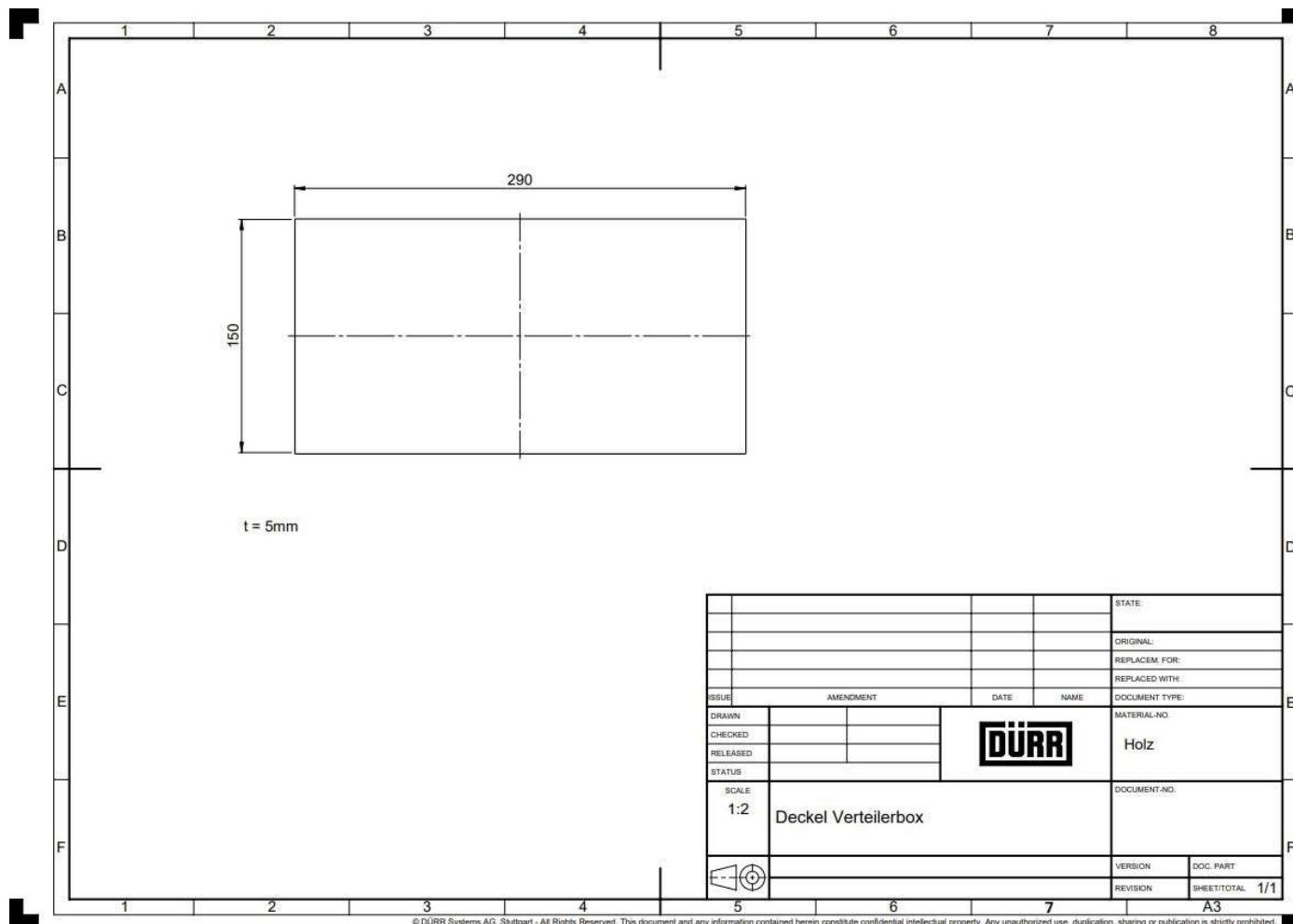
Griffstück



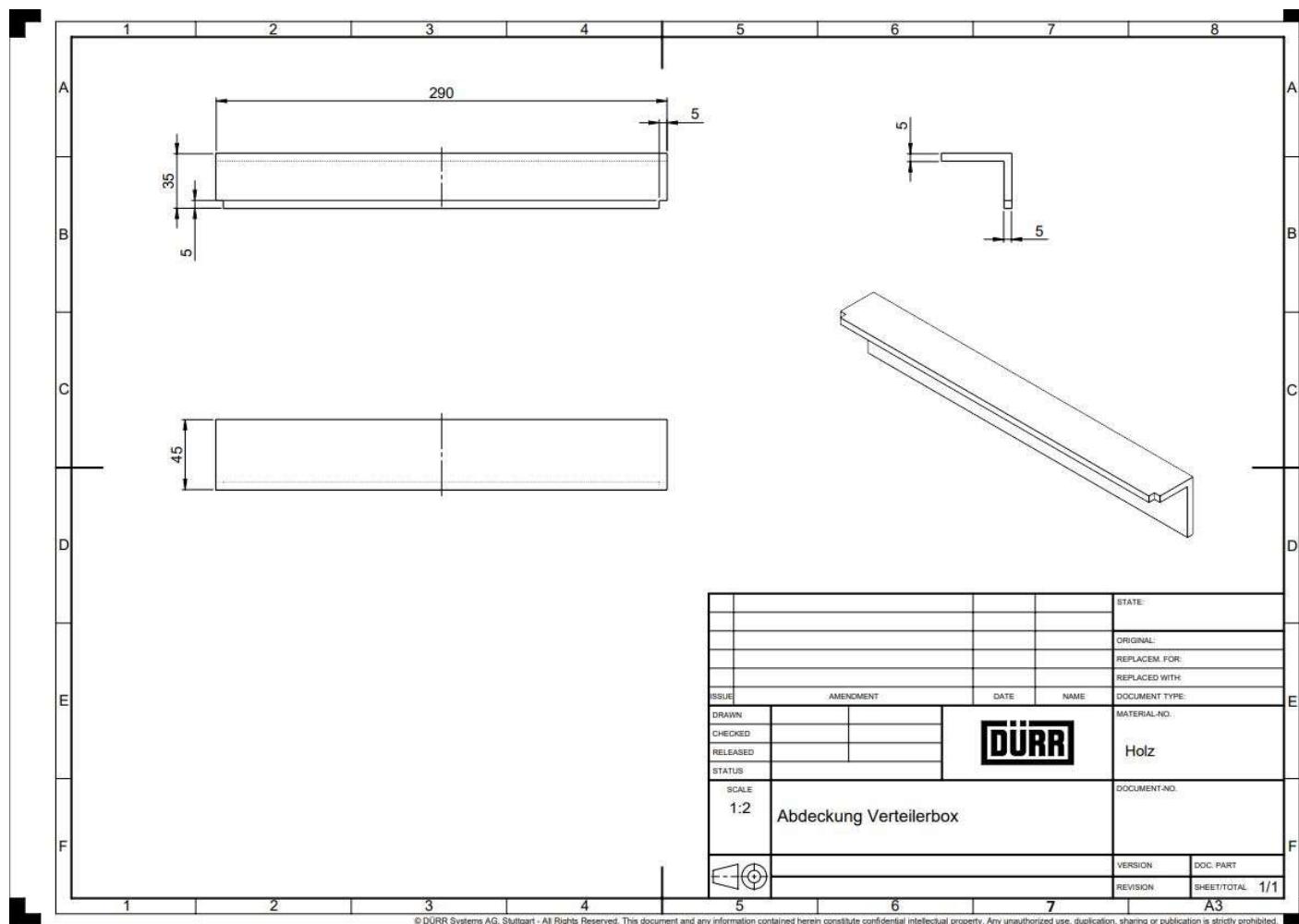
Distanzstück Griff



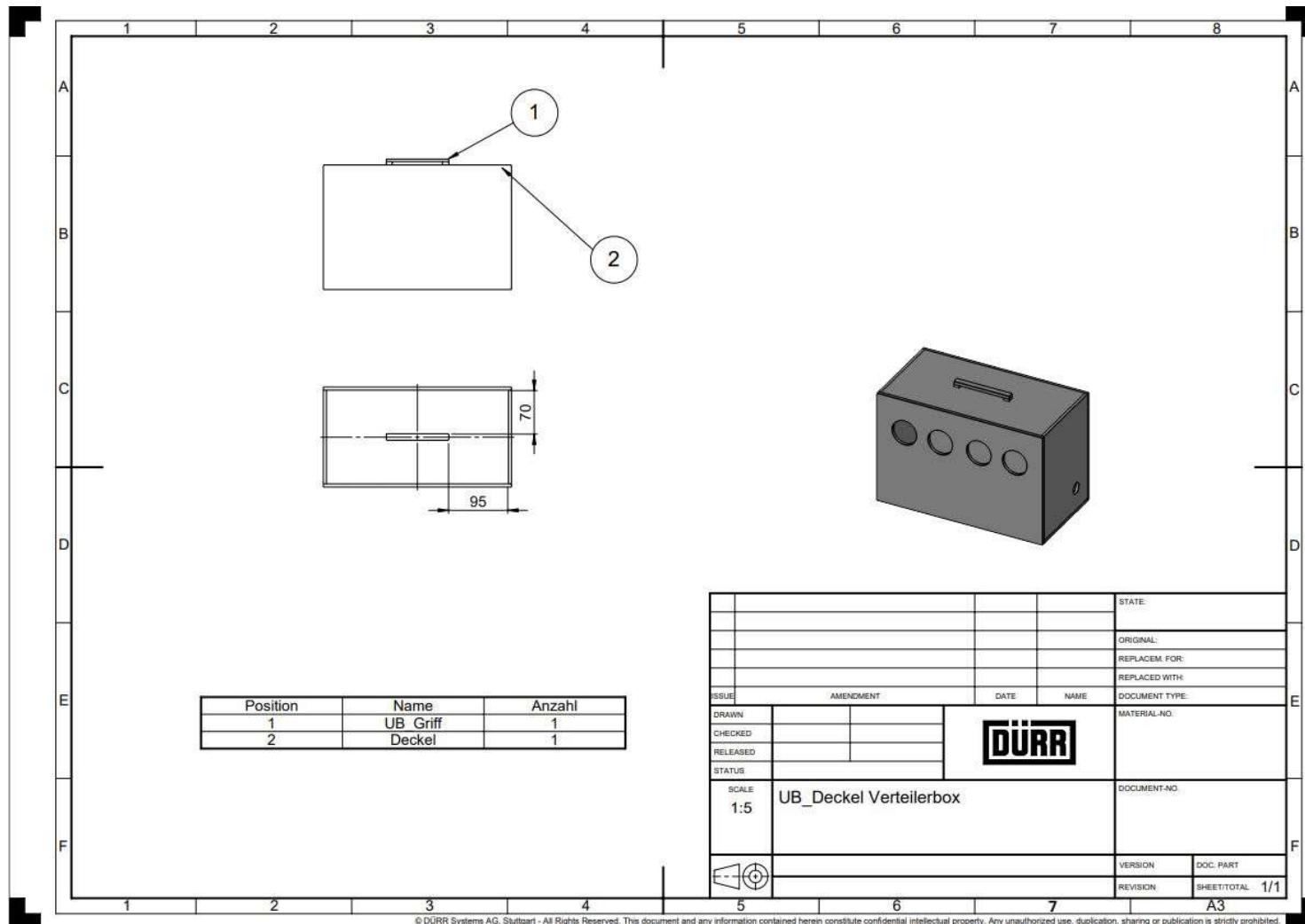
Deckel



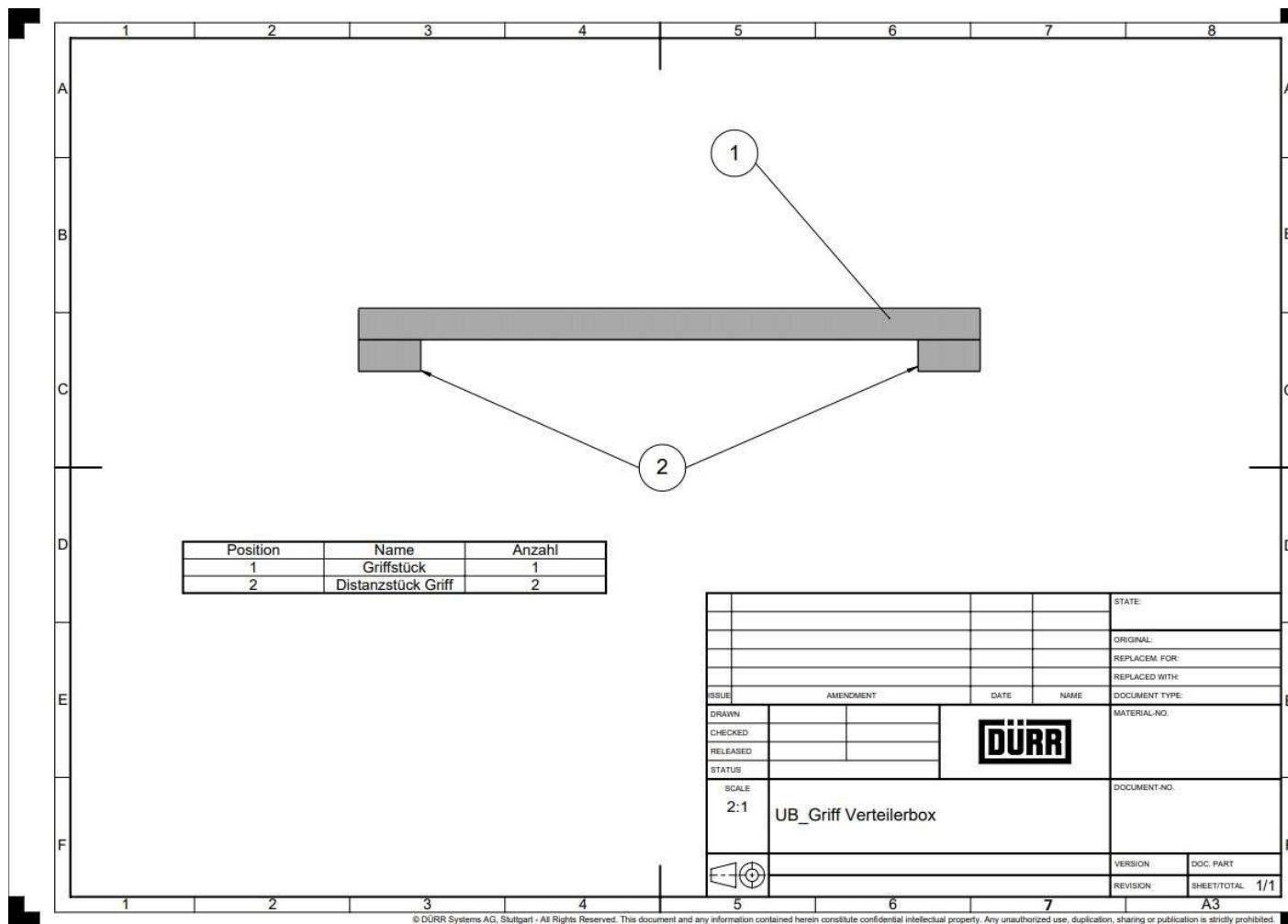
Schutzabdeckung



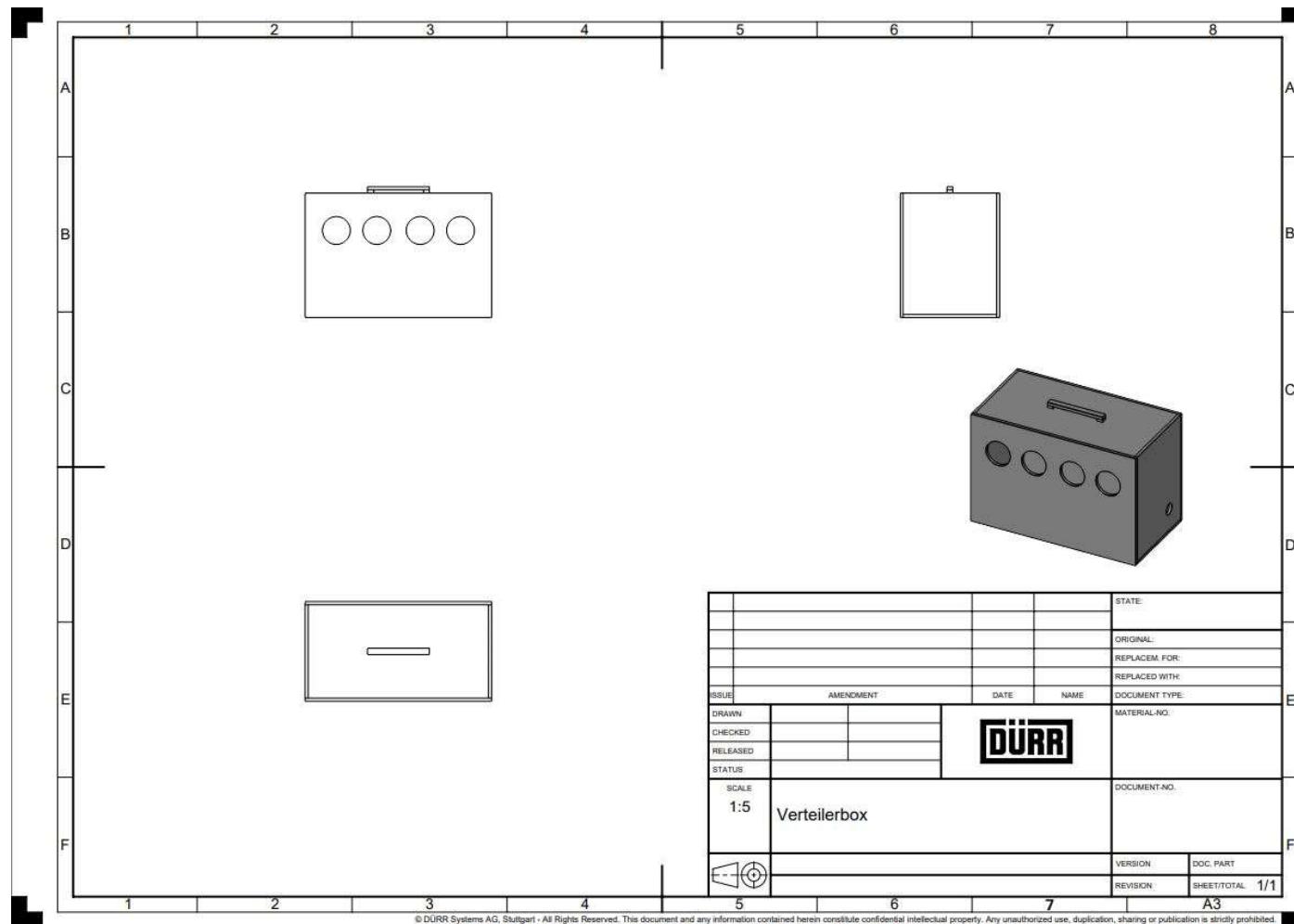
Unterbaugruppe Deckel



Unterbaugruppe Griff

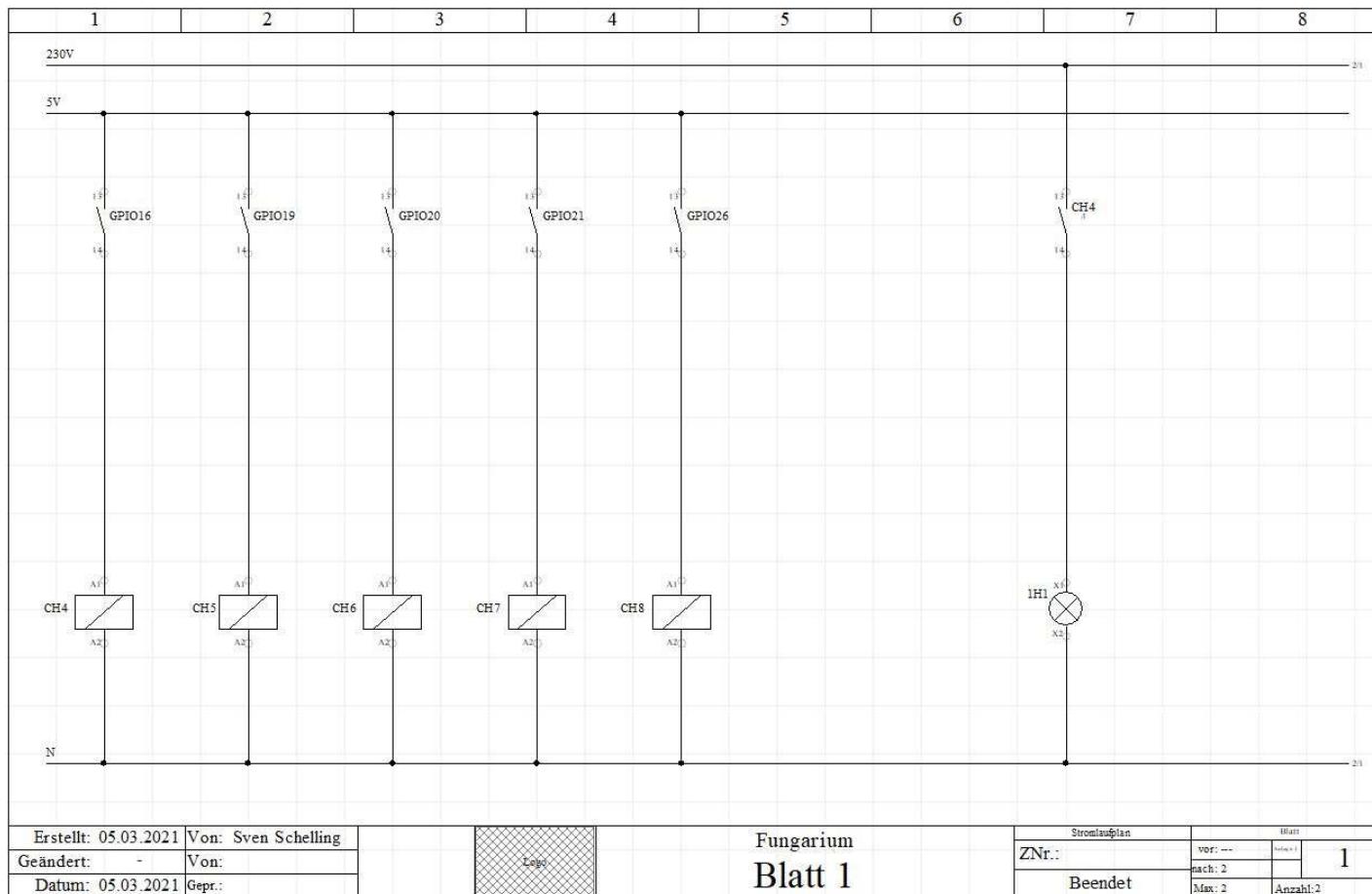


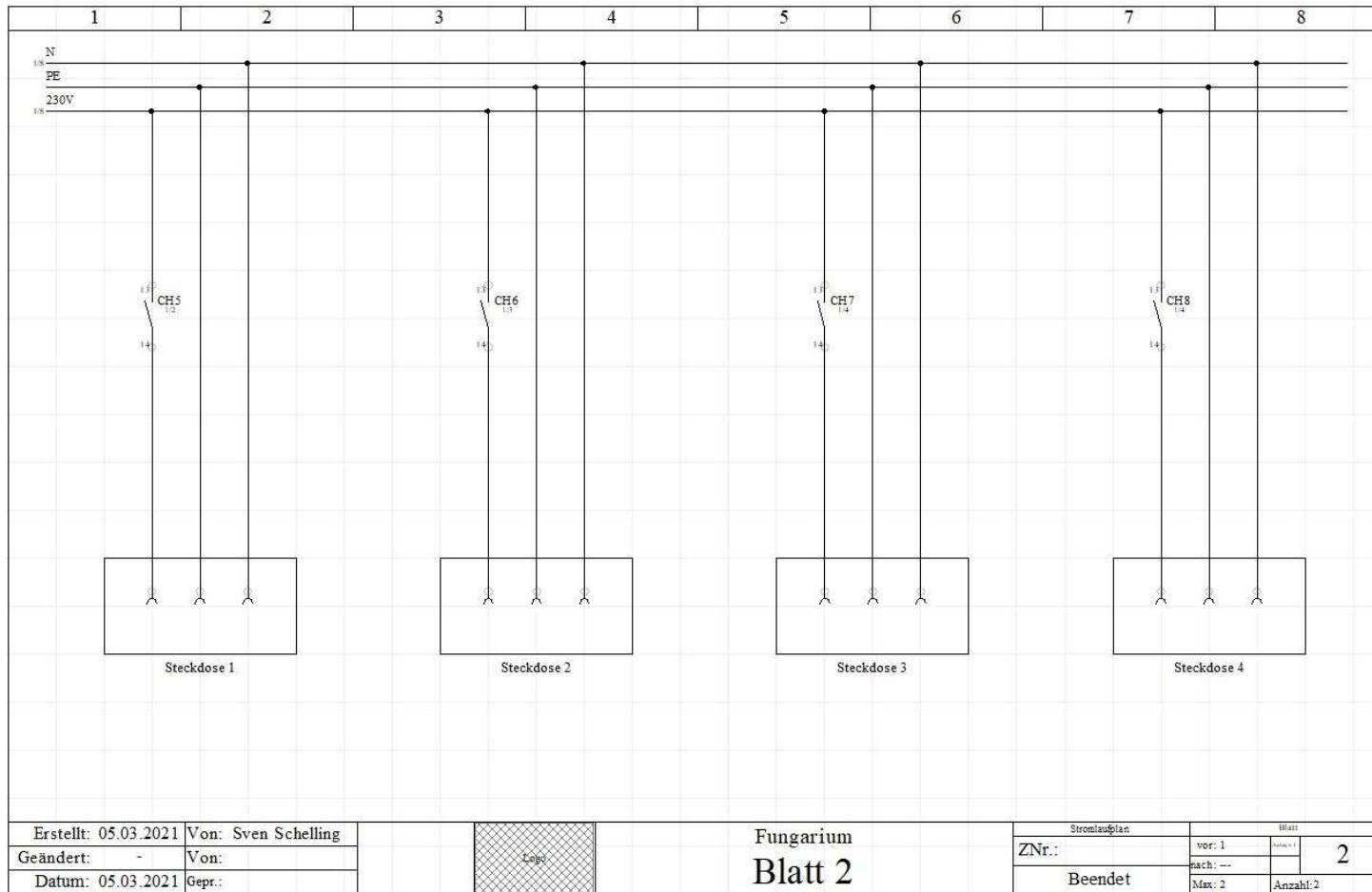
Zusammenbau



13.5 Elektrischer Schaltplan

Es ist an dieser Stelle anzumerken, dass es sich beim dargestellten Schaltplan um eine sehr vereinfachte Darstellung (nicht normgerecht). Dabei sind die einzelnen GPIO Pins, welche zum Ansteuern der Relais verwendet werden, durch einfache Schalter dargestellt.





Schaltplan des Relaisboard

