

Masterarbeit
im Studiengang
Mobile Systeme (MOS)

Konzeption und Implementierung eines Inventarisierungssystem in einer Laborumgebung

Referent : Prof. Dr. Elmar Cochlovius
Koreferent : Judith Jakob
Vorgelegt am : 23. August 2018
Vorgelegt von : Helge Meiering
Matrikelnummer: 256010
Elzstraße 1, 79297 Winden im Elztal
h.meiering@hs-furtwangen.de

Vorwort

Die folgende Arbeit wäre ohne die Unterstützung von Freunden, Familie und Betreuen nicht möglich gewesen. Von allen Seiten wurde ich ermuntert oder bekam hilfreiche Tipps. Dafür möchte ich mich herzlichst bedanken. Ich benutze häufiger für viele Wörter den maskulinen Ausdruck. Das mache ich der Einfachheit halber. Damit möchte ich kein Geschlecht unterdrücken oder benachteiligen und bitte daher um Verständnis.

Abstract

The present work shows how an inventory system can look like in a laboratory environment. A lot of importance is attached to the conception. In this phase, it is not only decided how the application should look at the end, but also how it is implemented. The knowledge of the developer has to be considered. It is necessary to measure the amount of learning required to familiarize yourself with the new and unknown technologies. At the end, an implementation must take place, which presents many challenges.

Die vorliegende Arbeit zeigt wie sich ein Inventarsystem in einer Laborumgebung aussehen kann. Dabei wird ein großer Wert auf die Konzeption gelegt. In dieser Phase entscheidet sich nicht nur wie die Anwendung zum Schluss aussehen soll, sondern auch wie sie umgesetzt wird. Es muss das Wissen vom Entwickler berücksichtigt werden. Dabei muss der Lernaufwand bemessen werden, welcher benötigt wird um sich in die neuen, noch unbekannten Technologien einzuarbeiten. Zum Schluss muss eine Umsetzung erfolgen, welche viele Herausforderungen stellt.

Inhaltsverzeichnis

Vorwort	i
Abstract	iii
Inhaltsverzeichnis	vii
Abbildungsverzeichnis	ix
Abkürzungsverzeichnis	xi
1 Einleitung	1
1.1 Problemstellung	1
1.2 Ziel der Arbeit	2
2 Vorgehen	3
2.1 Versionsverwaltung	3
2.2 Scrum	3
2.2.1 Scrum als Einzelperson	4
3 Konzeption	5
3.1 Analyse des aktuellen Webauftritts	5
3.1.1 Corporate Design	6
3.1.2 Erweiterung des Webauftritts	6
3.2 Konzeption des Inventarisierungssystem	8
3.3 Vergleichbare Inventarisierungssysteme	9
3.3.1 Magento	9
3.3.2 WooCommerce	10
3.3.3 Entscheidung zur eigenen Anwendung	10
3.4 Anforderungen	11
3.5 Datenbank	12
3.6 MockUps	13
3.7 User Experience	14
3.8 Frameworks	14

3.9	Wahl der Frameworks	14
4	Umsetzung	17
4.1	Aufbau einer Testumgebung	17
4.1.1	Integration in das Netzwerk	18
4.1.2	MySQL	19
4.1.3	Nginx	20
4.2	XAMPP	20
4.3	Austausch von Daten	21
4.4	Frontend	21
4.4.1	Installation von Angular	22
4.4.2	Commandlineinterface von Angular	22
4.4.3	TypeScript	24
4.4.4	Module	25
4.4.5	Components	25
4.4.6	Services	26
4.4.7	Direktiven	27
4.4.8	Routing	28
4.5	Backend	28
4.5.1	Laravel	28
4.5.2	Installation von Laravel	29
4.5.3	Artisan	32
4.5.4	Controller	32
4.5.5	REST	33
4.6	Sicherheit	34
4.6.1	CORS	34
4.6.2	Token	34
4.6.3	Salt	35
4.7	Deploy	35
4.7.1	Deploy in Angular	36
4.7.2	Deploy in Laravel	36
5	Anwendung	37
5.1	Anlegen, Aktualisieren, Löschen und Ausgeben von Gegenständen . . .	37

5.2	Ausgabe von allen Gegenständen	40
5.3	Routing Strategie	41
5.4	E-Mail Reminder	41
5.5	Datenbank import und export	42
5.6	Gestalten der Navigationsleiste	45
6	Diskussion der Ergebnisse	47
6.1	Fazit	47
6.2	Ausblick	47
	Eidesstattliche Erklärung	49
A	Anhang	51

Abbildungsverzeichnis

Abbildung 1: Ablauf eines Scrums. [Niermann.2017]	4
Abbildung 2: Die Startseite des Webauftritts	5
Abbildung 3: Raumkarte	7
Abbildung 4: Netzwerkkarte	8
Abbildung 5: Magento Itemverwaltung	9
Abbildung 6: ER-Diagramm	12
Abbildung 7: MockUp von Items-Tabelle	13
Abbildung 8: Button Gruppierung	14
Abbildung 9: VM-Ware mit geladenem Abbild	17
Abbildung 10: Netzwerkeigenschaften im VM-Ware-Player	19
Abbildung 11: WinSCP	21
Abbildung 12: Projektordner: Angularprojekt	23
Abbildung 13: Sourceordner in Angular Projekt	23
Abbildung 14: App Ordner in Src-Ordner des Angularprojektes	23
Abbildung 15: Dateien einer Component	26
Abbildung 16: Dateien eines Services	27
Abbildung 17: Dateien einer Direktiven	28
Abbildung 18: Root Ordner einer Lavelanwendung	31
Abbildung 19: Ergebnis von einer REST-URL	34
Abbildung 20: Alle User nutzen das Passwort "test"	35
Abbildung 21: Anlegen von Items	40
Abbildung 22: Importieren und Exportieren von Items	45

Abkürzungsverzeichnis

bzw. beziehungsweise

CD Corporate Design

CLI Command Line Interface

CMS Content Management System

CORS Cross-Origin Resource Sharing

CSS Cascading Style Sheets

Ecma Ecma International

FTP File Transfere Protocol

GNU General Public License

GPL General Public License

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

IDE integrated development environment

IP Internet Protocoll Adresse

JSON JavaScript Object Notation

MAC Media-Access-Control-Adresse

MIT Massachusetts Institute of Technology

MVC Modell View Controller

NPM Node Package Manager

REST Representational-State-Transfer

UI User-Interface

uvm. und vieles mehr

URL Uniform Resource Locator

UX User Experience

WLAN Wireless-Local-Area-Network

PDO PHP: Data Objects

PHP PHP: Hypertext Preprocessor

SASS Syntactically Awesome Stylesheets

SCP Secure Copy

SHL Smart Home Labor

SSH Secure Shell

SOP Same-Origin-Policy

SQL Structured Query Language

VM Virtuelle Maschine

XML Extensible Markup Language

z. Dt. zu Detusch

1 Einleitung

Das Smarthome-Labor der Hochschule Furtwangen umfasst vier einzelne Räume und ein Arbeitsbereich, in welchem sich Studierende, Mitarbeiterinnen und Mitarbeiter sich auf ihre wichtigen Arbeiten im Labor konzentrieren können. Das noch junge Labor steckt voller Leben. Es wird mit Bachelor-Projekten, Bachelor-Thesen und Master-Thesen immer weiter ausgebaut und die Studierenden lernen praxisnah den Umgang mit Geräten in einer Smarthome-Umgebung. Diese betrifft nicht nur die Unterkunft in welcher Personen leben, sondern auch die Industrie 4.0.

Im Sommersemester 2017 entstand die Masterthesis von Ashiq Mohamed Akbar Ali mit dem Titel "Creating a website with Info-Terminal and Live CCTV Stream for the Smart Home Laboratory at the Hochschule Furtwangen University". Mit dieser Arbeit wurde das Labor mit einer umfangreichen Webpräsenz ausgestattet, welche einen Einblick über die Vielfalt im Labor gibt. Das Labor wird hierbei vorgestellt und es wird auf einzelne Details eingegangen. [akbarali]

Es gibt eine Vielzahl an Geräten, Sensoren, Mikrocomputer und Alltagsgegenstände, wie zum Beispiel das Bett, mit welchen die Studierenden experimentieren und einen Betrag für die Wissenschaft und Öffentlichkeit bieten können.

1.1 Problemstellung

Durch diese große Anzahl von Geräten, Sensoren und Mikrocomputer entsteht auch eine Unübersichtlichkeit. Zwar wurden alle Gegenstände erfasst und Dokumentiert, jedoch ist das nur in einer Tabelle erfasst worden. Ein Durchsuchen dieser Tabelle kann Aufwendig sein wenn ein bestimmtes Gerät und dessen Status überprüft werden soll. Für ihre Arbeiten müssen sich Studenten auch Geräte reservieren oder gar ausleihen um sie in fremden Umgebungen testen zu können oder um daheim weiter arbeiten zu können. Dies in einer Tabelle zu erfassen ist nicht unmöglich, aber es handelt sich hierbei um einen großen Aufwand. Dieser gestaltet die Arbeit von einer Mitarbeiterin oder einem Mitarbeiter, als sehr Aufwendig. Diese Tabelle muss ständig kontrolliert und aktualisiert werden. Auch muss geprüft werden, ob die Leihe für ein Gerät schon verstrichen ist.

1.2 Ziel der Arbeit

Ziel ist eine Übersicht in das umfangreiche Inventar des Smarthome-Labors zu geben. Dabei kann eine Webanwendung helfen, welche in einer Tabelle Alle Geräte, Sensoren und Mikrocomputer aufnimmt. Diese Tabelle kann mittels Software schnell durchsucht und einfach erweitert werden. Auch soll die Mitarbeiterin oder der Mitarbeiter, wie auch Studierende an das Ende einer Leihfrist erinnert werden. Auch hierbei ist die Software eine Lösung, indem sie Erinnerungsmails verschickt.

2 Vorgehen

In der Vorbereitung wurden schon früh die Werkzeuge für die Umsetzung und Konzeption gewählt, welche das Projekt strukturieren und planen sollten. Dabei kamen Mittel wie Kugelschreiber, Notizbücher, Kalender oder auch die Mindmap zum Einsatz. Mittels Requirement Engineering wurde eine Anforderungsanalyse erstellt. Um einen übersichtlichen Workflow zu haben, fiel die Entscheidung Scrum als Vorgehensmodell für den künftigen Ablauf zu wählen.

2.1 Versionsverwaltung

Für die Versionsverwaltung wurde Git gewählt. Es handelt sich um ein Dezentrales Versionsverwaltungssystem und ist nicht von einem bestimmten Server verbunden. In einem Versionsverwaltungssystem werden verschiedene Entwicklungszustände gesichert. Damit ist es möglich zu sehen, wie sich eine Software entwickelt hat. Sollte ein Fehler auftreten, kann zu einer älteren Version gewechselt werden, in welcher der Fehler nicht vorhanden ist. Diese Versionen werden in einem Repository verwaltet. Viele Programmierprojekte liefern Daten, die Betriebssystem abhängig sind und nur unnötig für eine Versionierung sind. Für diesen Zweck kann eine `.gitignore` Datei angelegt werden. Diese Datei definiert welche Dateien von Git ignoriert werden. [git]

2.2 Scrum

Unter Scrum versteht man in der Projektplanung ein agiles Vorgehensmodell in der Softwareentwicklung. Es besteht aus mehreren Komponenten wie, Rollen Artefakte und Meetings. Sobald die Anforderungen und Eigenschaften eines Produktes angelegt worden sind, legt sie der Product Owner in einem sogenannten Product Backlog an. Daraufhin wird ein Sprint geplant. Es werden Anforderungen und Eigenschaften gewählt, welche in einem bestimmten Zeitraum erledigt werden können. Diese wiederum werden in das sogenannte Sprint Backlog abgelegt. Darauf erfolgt ein Sprint. Welcher nicht länger als ein Monat dauern sollte. Während des Sprints tauscht sich das Entwicklerteam über den täglichen Status der Entwicklung aus. Man spricht von einem Daily Meeting. Jeder ist auf dem aktuellen Stand und es kann sofort eingegriffen werden, wenn es an einer Stelle zu Schwierigkeiten kommt. Ist ein Sprint abgeschlossen kann ein funktionierender Softwareteil präsentiert werden. [Niermann.2017]

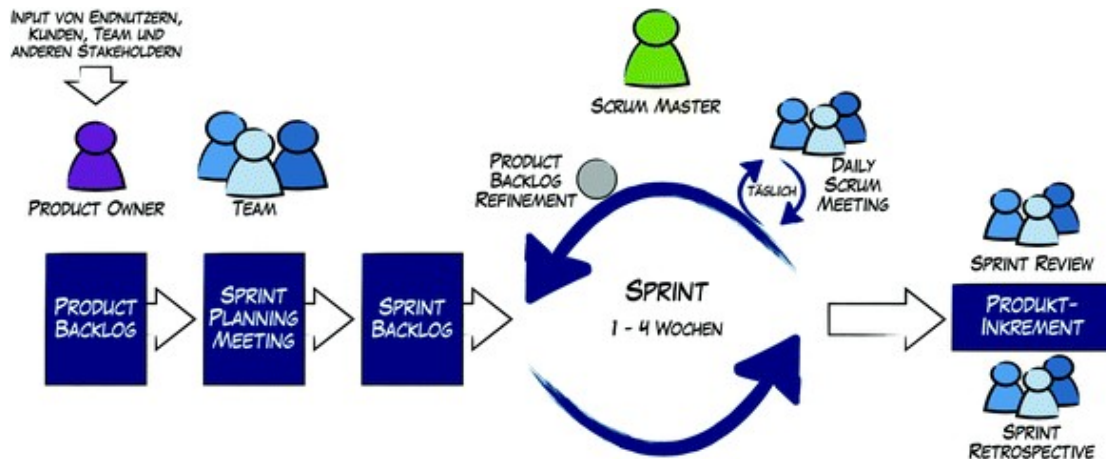


Abbildung 1: Ablauf eines Scrums. [Niermann.2017]

2.2.1 Scrum als Einzelperson

In dieser Arbeit wurde alleine und nicht im Team gearbeitet. Auch wenn es hierdurch keine Rollenverteilung gab, so konnte Scrum doch ideal eingesetzt werden. Das Backlog wurde mit Hilfe der Konzeption erzeugt. Mit der Planung wurde klar, welche Anforderungen an die Software bestehen. Diese wurden alle handschriftlich in ein Productbacklog niedergeschrieben. Darauf wurde mit dem Betreuer besprochen, welche Aufgaben innerhalb von zwei Wochen erledigt sein sollen und wurden, ebenfalls handschriftlich, in das Sprintbacklog aufgenommen. Während des Sprints sind die Daily Meetings entfallen, da diese nicht nötig waren. Als einzelne Person ist man immer über den aktuellen Stand seiner Entwicklung bewusst. Ein tägliches Austauschen mit Teammitgliedern ist nicht nötig, da es kein Team gibt. Nach den vergangenen zwei Wochen, wurde der Aktuelle Stand in einem Protokoll dokumentiert. Erfolge und Probleme wurden im Anschluss mit dem Erstbetreuer und der Zweitbetreuerin besprochen. Daraufhin wurde, wenn es nötig wurde, das Productbacklog erweitert. Aus diesem Backlog wurden erneut Aufgaben für den nächsten Sprint gewählt. Der Entwickler übernahm so alle Rollen, welche in dem Vorgehensmodell Scrum definiert sind.

3 Konzeption

Bei der Konzeption wurde der aktuelle Stand des bestehenden Webauftritts betrachtet und analysiert. Im weiteren Schritt wurden Erweiterungen für die Website geplant. Im nächsten großen Schritt erfolgte eine Analyse aktuell bestehender Software, welche Gegenstände verwaltet.

3.1 Analyse des aktuellen Webauftritts

Der Webauftritt, erreichbar im Netzwerk der Hochschule Furtwangen <http://web.smarthome.hs-furtwangen.de/>, des Smart Home Labor (SHL) Labors, wird mit der Hilfe des Content Management System (CMS) WordPress verwaltet. WordPress eigne sich hier, weil es mittels User Experience (UX) so gestaltet wurde, das es sehr leicht zu erlernen ist. Unerfahrene Benutzer könne so schnell eigene Inhalte einpflegen.

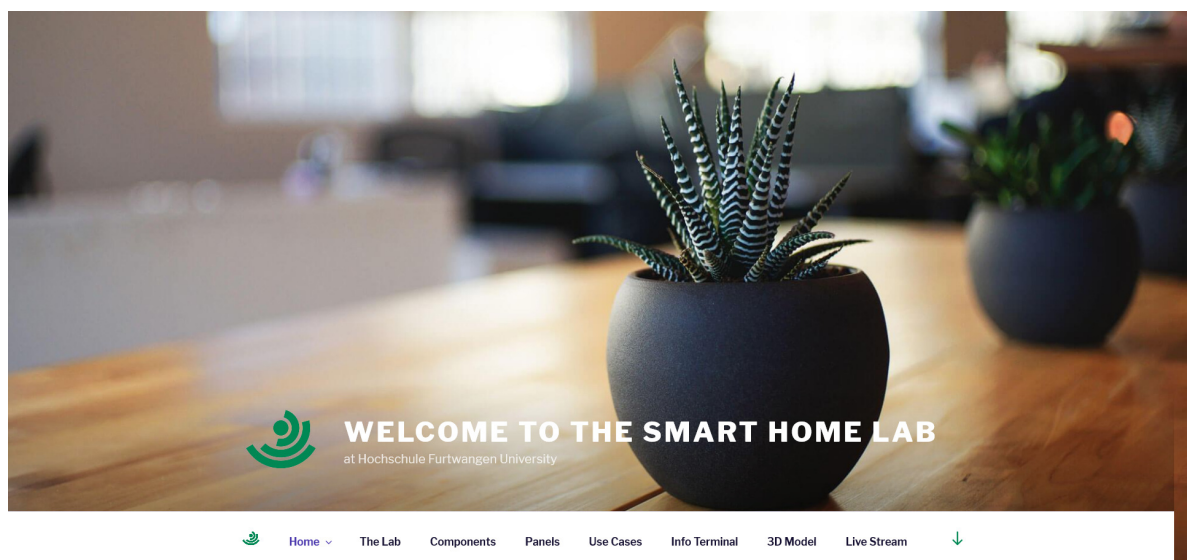


Abbildung 2: Die Startseite des Webauftritts

Die Website ist mit dem, von WordPress eigenem Designe Twentyseventeen gestaltet. Auf der Startseite sieht man ein großes Headerbild. Darunter folgt eine Navigationsleiste, welche die Links bei einem Besuch einer Unterseite in einem Lila Farbton darstellt. Die Navigation umfasst die Seiten:

- Home
- The Lab

- Components
- Panels
- Use Cases
- Info Terminal
- 3D Model
- Live Stream

Weiter folgt eine kurze Beschreibung über das Labor. Der nächste Abschnitt der Startseite stellt das Team vor. Im Anschluss sieht man noch ein Video, welches das Labor vorstellt, ein Kontaktformular und die Adresse mit einer Google Maps Karte. Am untersten Ende ist ein Footer, welche eine Copyright und einen Link zum Impressum enthält.

Die Unterseite „The Lab“ stellt das Labor mit Grundrissen etwas detailreicher vor. „Components“ stellt wenige, aber wichtige Geräte vor. In „Panels“ werden, die mit Sensoren und Geräte versehenen, Wände in den Räumen beschrieben. Eine Auswahl von umgesetzten Anwendungsfällen werden in der Unterseite „Use Cases“ beschrieben. Für das Infoterminal gibt es ebenfalls eine Unterseite, welche automatisiert eine Präsentation über das Labor abspielt. Unter dem Punkt „3D Model“, befindet sich eine 3D Ansicht des Labors, welches sich in einer Ego- und Vogelperspektive betrachten lässt. Auch können hier vereinzelt Geräte bedient werden. Abgeschlossen wird die Navigation mit einer Unterseite, welche einen Video Livestream zeigen kann.

3.1.1 Corporate Design

Mit den aktuellen Farben entspricht der Webaufttritt des Labors nicht dem Corporate Design (CD) der Hochschule Furtwangen. So müssen die Lila Farben durch die Farbe Grün mit dem Hexwert 83b62d geändert werden. Dies ist wichtig um einen Wiedererkennungswert der Hochschule darzustellen. Die Navigation stellte hier einen größeren Bruch dar. Um die Seite vollständig im CD der Hochschule zu haben, ist ein Expertengespräch nötig.

3.1.2 Erweiterung des Webaufttritts

Neben dem Corporate Design (CD) gab es noch weitere Punkte, welche den Auftritt noch Informativer und attraktiver gestalten konnten. So stehen im Labor die einzelnen Räume: Küche, Bad, Multimediarraum, IoT-Raum und der Arbeitsbereich stark im Vordergrund. Diese wurden bisher nur sehr dürftig auf der Homepage erwähnt. Daher wurde für die Räume eigene Unterseiten angelegt. Um auf User Experience (UX) zu

achten, wurde mittels des WordPress-Plugin Draw Attention eine Interaktive Karte erzeugt. Bei einem Klick auf einen Raum, erhält der User mehr Informationen. [WPDDrawAttention.]

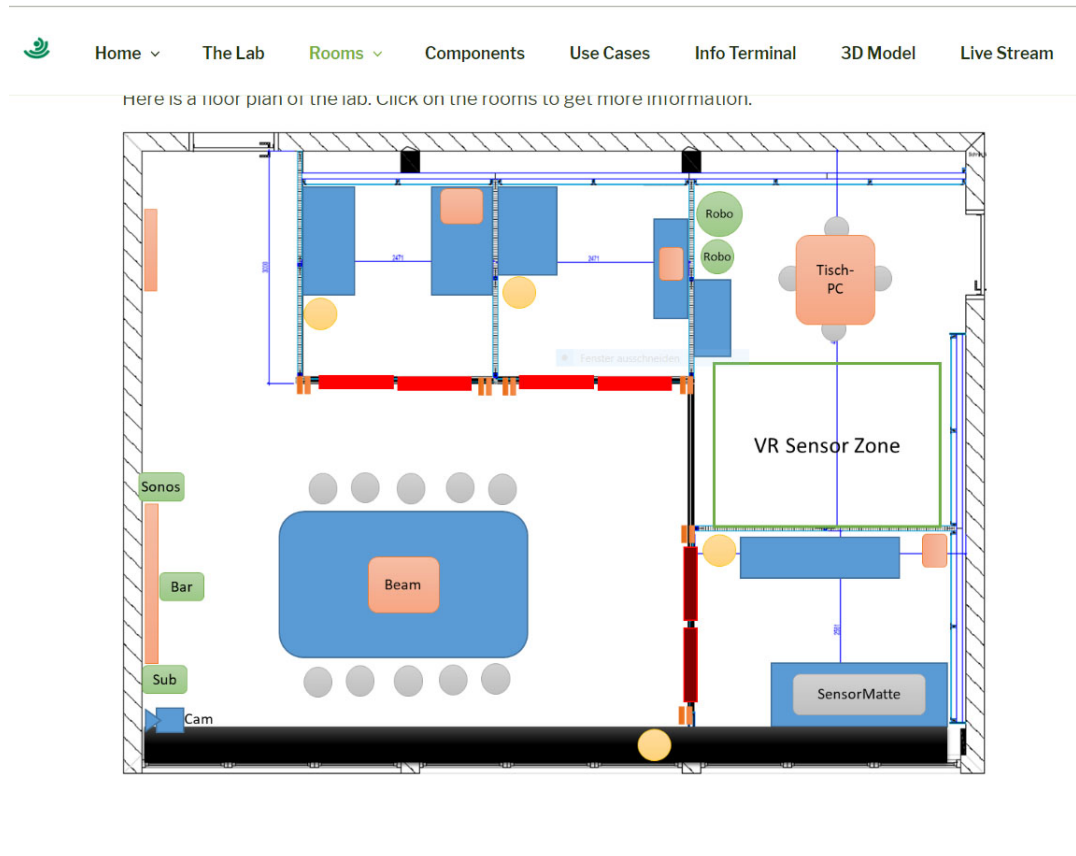
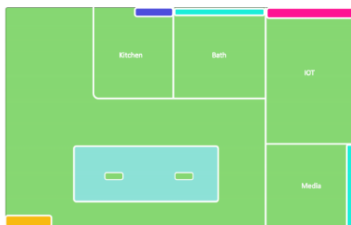


Abbildung 3: Raumkarte

Jeder Raum hat seine eigene Unterseite, welche die Aufgaben des Raumes und dessen Ausstattung beschreibt.

Die Netzwerkarchitektur im Labor ist sehr komplex. Wenn man nicht gerade mit am Aufbau dieser Architektur gearbeitet hat, kann schnell der Überblick verloren werden. Eine Übersichtskarte (siehe Abbildung 4), auf welcher man sieht wo sich Server, Router und Gateways befinden, kann helfen um herauszufinden, wo bei Problemen mal nachgesehen werden kann. Diese findet man in der Navigation unter dem neuen Punkt Räume. Neben der Karte gibt es zu jedem Raum eine eigene Unterseite, wo auf dessen Netzwerkeigenschaften genauer eingegangen wird. Die Karte wurde mit Adobe Illustrator CC 2017 erzeugt. Dabei wurde, um die Übersichtlichkeit nicht zu gefährden, sehr auf Schlichtheit geachtet. Nur das nötigste wurde eingezeichnet. Neben der Einfachheit halten sich die Farben Grün und Weiß an das CD der Hochschule Furtwangen.



On the graphic you can see the different rooms of the laboratory. The network runs around the lab using a network cable. Thus, in each room with the help of LAN, devices can be integrated into the network. There are also two WLANs available. So also wireless devices can be integrated.

Network in:

- IOT room
- Bath
- Kitchen
- Media room
- Workplace

Abbildung 4: Netzwerkkarte

3.2 Konzeption des Inventarisierungssystem

Das Inventarisierungssystem ist das Herzstück dieser Masterthesis. Mit ihm können alle Gegenstände verwaltet werden und dazu noch sehr einfach und übersichtlich. Dabei spielt User Experience (UX) und damit auch das User-Interface (UI) eine sehr große Rolle. Um einen Überblick zu bekommen was der aktuelle Markt zu bieten hat, müssen Vergleichbare Verwaltungssysteme(siehe Kapitel 3.3) analysiert werden. Damit eine reibungslose Programmierung erfolgen kann müssen Konzepte für die Architektur der Anwendung ausgearbeitet werden.

3.3 Vergleichbare Inventarisierungssysteme

Sucht man nach Inventarisierungssystemen stößt man immer wieder auf Software für Webshops. Mit ihnen hat man sehr häufig einen riesigen Umfang an Funktionen, mit welchen man nicht nur sein Inventar, sondern auch seine Verkäufe verwalten kann.

3.3.1 Magento

Magento¹ ist eine Openen-Source-E-Commerce Plattform und steht unter der Open Software License². Umgesetzt wurde es mit dem PHP: Hypertext Preprocessors (PHPs) Framework Zend und lässt sich durch zahlreiche Plug-Ins erweitern.[.2018]

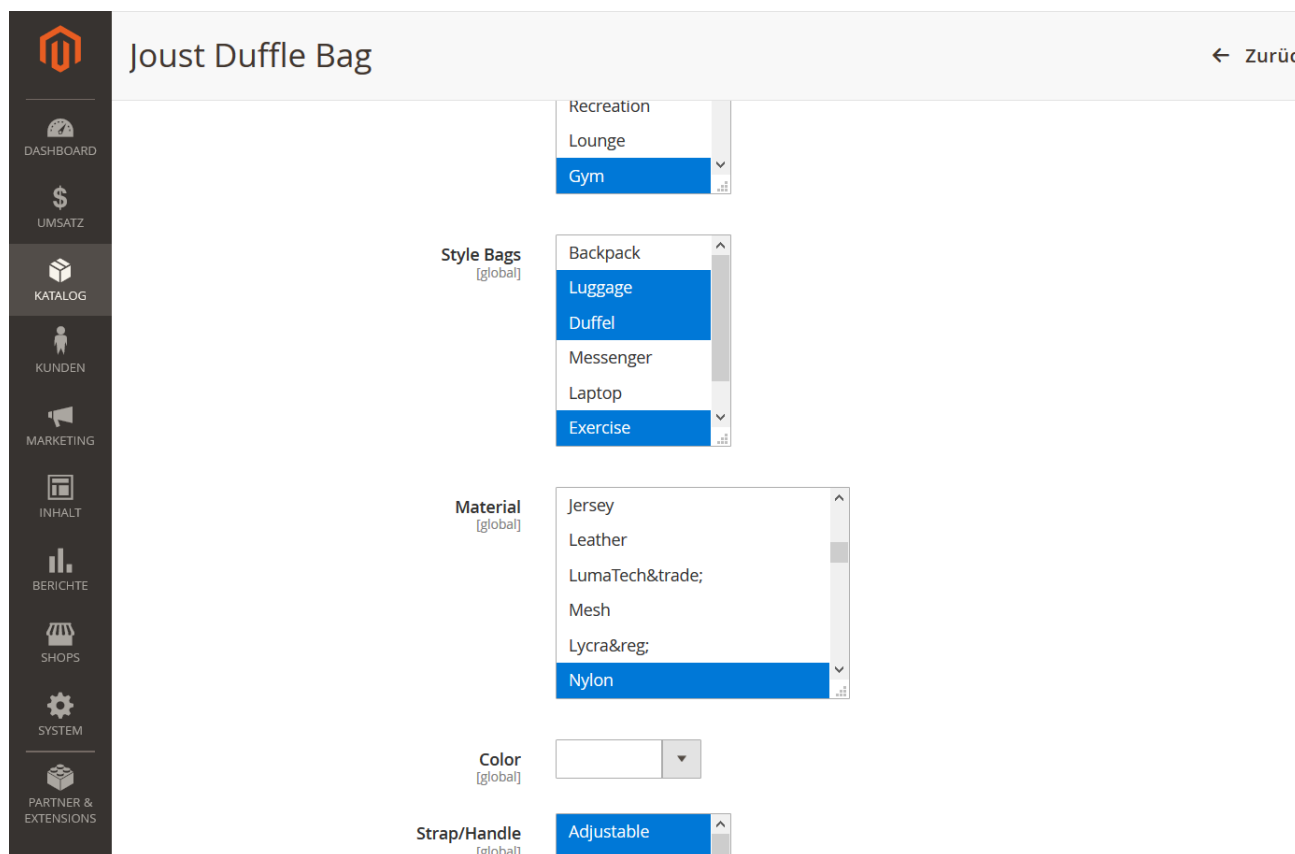


Abbildung 5: Magento Itemverwaltung

3.3.1.1 Vorteile von Magento

Magento bringt zahlreiche Verwaltungsoptionen. Man kann bis in das kleinste Detail einen Webshop konfigurieren und verwalten. Eine Rollenverteilung der User wird ebenfalls geboten. Jeder Gegenstand kann mit vorgefertigten und eigen angelegten Eigenschaften versehen werden. Das UI von Magento ist ähnlich wie bei WordPress

¹<https://magento.com>

²<https://opensource.org/licenses/osl-3.0.php>

sehr benutzerfreundlich. Mit ein wenig Erfahrung findet man sich schnell zurecht. [Koch.2012]

3.3.1.2 Nachteile von Magento

In seiner Größe, liegt der Nachteil des Frameworks. Es bietet zu viele Funktionen, welche nicht für ein Inventursystem in einer Laborumgebung benötigt werden. Damit belegt es unnötig Speicherplatz und lädt Dateien, welche nicht gebraucht werden. Auch wenn die Lernkurve sehr flach ist, muss eine gewisse Einarbeitung doch erfolgen. Auch ist mit der aktuellen Version von Magento eine Aktualisierung der PHP Version nötig. [Koch.2012]

3.3.2 WooCommerce

Anders als bei Magento (siehe Kapitel 3.3.1) steht das WordPress-Plugin, welches auch als E-Commerce Software dient unter der General Public License (GNU) Lizenz.

3.3.2.1 Vorteile von WooCommerce

Bei kostenlose WordPress-Plugin ist kostenlos und sehr einfach zu installieren. Mit einem Klick ist es aus dem Plugin Bereich gewählt und kann mit einem Installationsassistenten den eigenen Bedürfnissen nach angepasst werden. Ein erfahrener WordPress-Nutzer hat eine deutlich niedrigere Lernkurve, als bei Magento [WooCommerce.2018]

3.3.2.2 Nachteile von WooCommerce

Die Nachteile sind ähnlich wie bei Magento. Es beinhaltet zu viele Funktionen, welche nicht gebraucht werden. Außerdem ist es sehr schwer möglich es nicht als Webshop, sondern als Verwaltungssystem zu nutzen. WooCommerce gibt hier sehr starke Richtlinien vor. So muss man schon im Installationsassistenten sich Gedanken über Maße, Bezahlmethoden und Versand machen. Was für ein Inventarisierungssystem nicht nötig ist. Mit WooCommerce ist man stark an WordPress gebunden. Sollte man das CMS wechseln wollen, ist ein mitnehmen der Anwendung nicht möglich.

3.3.3 Entscheidung zur eigenen Anwendung

Die Entscheidung eine eigene Anwendung zu schreiben wurde schon früh getroffen. Es muss eine Anwendung zur Verfügung gestellt werden, welche nicht mit Funktionen überladen ist. Die Lernkurve muss für jeden Anwender möglichst flach gehalten sein. Der Aufwand Magento oder WooCommerce in das Hochschul CD zu bringen, kann ein großer Aufwand sein. Daher lohnt es sich die Anwendung selbst zu schreiben. So bleibt sie konfigurierbar und es werden nur PHP Kenntnisse und keine speziellen WordPress

oder Magento-Kenntnisse vorausgesetzt. Bei einem Wechsel zu einem anderen CMS kann die Anwendung ebenfalls übernommen werden.

3.4 Anforderungen

Mit dem Anforderungsmanagement werden mögliche Features definiert. In der Anforderungsanalyse werden Eigenschaften, Funktionalitäten und die Qualität an die Software Festgehalten. [Grande.2014] Es werden funktionale und nicht-funktionale Anforderungen niedergeschrieben. Mit diesen Anforderung entsteht eine Definition und Ziele an die Anwendung.

- Das Inventarisierungssystem muss alle Items in einer Laborumgebung aufnehmen.
- Dabei soll bekannt sein wie der aktuelle Zustand der jeweiligen Items ist.
- Die Datenbank sollen durchsuchbar sein.
- Es müssen neue Items angelegt und alte gelöscht werden können.
- Es soll für jeden Benutzer einen persönlichen Bereich geben.
- Es soll Verschiedene Rollen für die Administration und User geben.
- Jeder Benutzer muss seine Persönlichen Daten verändern können.
- Benutzer können über den Administrator oder einem dazu befugten Mitarbeiter Items leihen.
- Benutzer sollen an die Rückgabe der geliehenen Items erinnert werden.

Ist eine solche Liste von Anforderungen erstellt worden, erhält man eine gute Übersicht über die zu entwickelnden Teilbereiche.

3.5 Datenbank

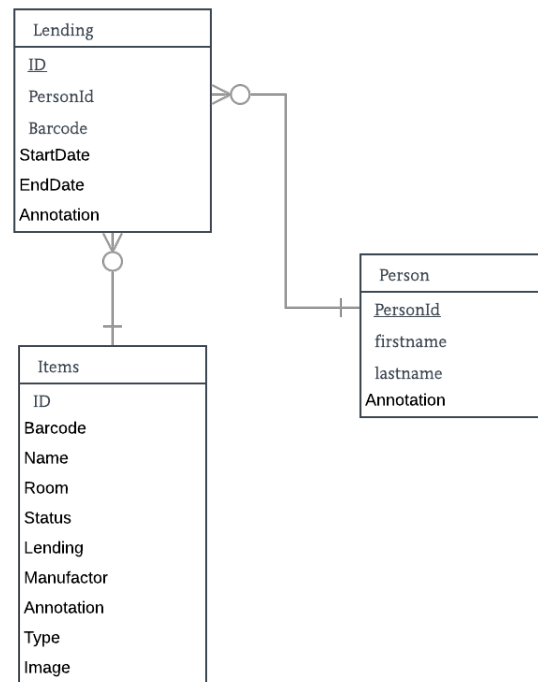


Abbildung 6: ER-Diagramm

Mit der Idee der Webanwendung, wurde schon sehr früh mittels einer Mindmap jede mögliche Eigenschaften aufgeschrieben. Aus dieser Mindmap konnte sehr komfortabel ein ER-Diagramm erzeugt. Um Redundanzen zu vermeiden, muss eine Normalisierung durchgeführt werden. In der Regel sind Redundanzen mit den ersten drei Normalformen abgeschlossen. Im ersten Schritt wurde darauf geachtet, dass alle Eigenschaften Atomar sind. Vorname, in der Datenbank firstname und Nachname, in der Datenbank lastname, müssen als einzelne Eigenschaften zur Verfügung stehen. Eine Eigenschaft, mit dem Namen fullname, ist nach der ersten Normalform nicht erlaubt. Um die zweite Normalform zu erfüllen, muss die erste Normalform erfüllt sein und jedes Nichtschlüsselattribut von jedem Schlüsselkandidaten voll funktional abhängig sein. So wird festgestellt welche Eigenschaften eindeutig sind. Das ist nicht immer möglich, wie auch im Fall der Items Tabelle. Jeder Barcode in der Items-Tabelle ist einzigartig, jedoch gibt es Produkte, welche bewusst keinen Barcode haben. Gelöst wurde das Problem mit dem Künstlichen Primärschlüssel ID.[**DatenbankenVerstehen.2018b**] Die höchste Priorität der Normalisierung ist das Vermeiden von Redundanzen und kann nur mit der 3. Normalform erreicht werden. Hierfür wurde die Hilfstabelle Lending eingefügt. Feststellen kann man das, in dem man überprüft, ob es viele-zu-viele existieren. Denn ohne diese Tabelle käme es zu Problemen, wenn sich eine Person mehrere Items leihen möchte.[**DatenbankenVerstehen.2018**]

3.6 MockUps

Um die geplante Idee umzusetzen ist es nötig ein MockUp zu erstellen. Doch bevor das MockUp erstellt werden kann, muss Storytelling betrieben werden. Beim Storytelling werden die Anforderungen in realistische kleine Geschichten erzählt. So wird es leichter sich vorzustellen, wie ein Programm funktioniert. Diese Geschichten werden sachlich aufgeschrieben. An diesen Geschichten kann früh festgestellt werden, ob sich eine Idee nicht oder nur sehr aufwändig umsetzen lässt. Hier tritt auch die User Experience in den Vordergrund, da man sich beim Schreiben in den User hineinversetzen muss. [Schach.2017]

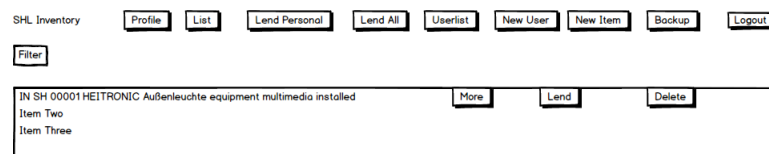


Abbildung 7: MockUp von Items-Tabelle

Ist das Storytelling abgeschlossen und die möglichen Fehler beseitigt wurden, so kann das MockUp beginnen. Beim MockUp, in der Webentwicklung, wird das Layout schemenhaft mit Bleistift und Papier erzeugt. Dabei tritt das Design stark in den Hintergrund, es ist in erster Linie wichtig, das verstanden wird, wie die Anwendung mit Button, Formulare und weiteren Elemente funktionieren. Das MockUp macht es später für die Entwicklung leichter, wenn es um die Umsetzung der Webanwendung geht. Bei der Erstellung des MockUp's hat man die Programmierung immer im Hinterkopf. Dabei wird überlegt wie Elemente angelegt werden.

3.7 User Experience

User Experience (UX) wurde häufiger in dieser Ausarbeitung verwendet und wurde noch nicht genauer erklärt. Das geschieht nun. Bei der User Experience, auf deutsch *Benutzererfahrung*, spricht man davon, wie der Benutzer mit Software Umgeht. Dabei greift man auf eigene Erfahrungen zurück, es gibt allerdings auch Werkzeuge um eine gute Benutzbarkeit zu messen. [Moser.2012] Dazu gehört Beispielsweise das Storyboard, welches mit hilfe des Storytelling Geschichten mit Bilder erzählt. Für dieses Projekt wurde auf kurze Wege geachtet, das der User die Maus nicht allzu sehr die Maus bewegen. Auch wird dem User durch das Designe gezeigt was klickbar ist und was nicht. Unbewusst wird er eine Gruppierung vornehmen. So sind Buttons, mit welchen etwas erzeugt wird, wie *New Item*, *New User* und *Backup*, mit einem anderen Hintergrund und einer Umrandung versehen.



Abbildung 8: Button Gruppierung

3.8 Frameworks

Wenn man Framework in die deutsche Sprache übersetzt, so erhält man das Wort Rahmengerüst. In der Softwareentwicklung spricht man von solchen Frameworks bei Programmierstrukturen, welche den Entwickler unterstützen. Dabei handelt es sich nicht um eine fertige Software. Es handelt sich um bewährte Programmierstrukturen, welche dem Entwickler die Arbeit abnehmen. So werden bekannte Entwurfsmuster übernommen werden. Auch aufwendige Funktionen sind in Frameworks schon häufig implementiert. Der Entwickler muss sich nicht mehr aufwendig um eine Datenverbindung kümmern oder kann komplizierte Animationen mit wenigen Zeilen Code erstellen. Eine wichtige Aufgabe während der Softwareentwicklung ist das Testen. Hier werden mithilfe des Frameworks Tests geschrieben, mit welchen kontrolliert werden kann, ob die Software korrekt funktioniert. Frameworks kommen meistens in der Objektorientierung zum Einsatz. Ihre Einsatzgebiete können vielseitig sein. Man findet sie in der Softwareentwicklung, bei der Entwicklung von Spielen und in der Webentwicklung.

3.9 Wahl der Frameworks

Bei der Wahl der Frameworks wurde analysiert welches sicher läuft, eine große und aktive Community hat, wenig Einrichtungen erfordern und in welchen die meisten Erfahrungen schon vorhanden sind. Bei der Wahl des Frontend-Frameworks fiel die Wahl

schon früh auf Angular. Es wird bis heute noch von Google weiterentwickelt und hat eine sehr große Community. Die Erfahrungen in JavaScript sind ebenfalls vorhanden und die meisten Einrichtungen sind schon vorhanden oder werden von dem Framework selbst übernommen. Bei der Wahl des Backend-Frameworks war die Wahl nicht so einfach. Es standen Tomcat, ein Java-Framework und Laravel, ein PHP-Framework, zur Auswahl. Es wurde Laravel gewählt, da es von sich aus viele Aufgaben, wie Sicherheit, Datenbankverwaltung oder Benutzerverwaltung mit Login-System übernimmt. PHP und MySQL waren auf dem Server installiert. Für Tomcat hätte eine separate Installation erfolgen müssen.

4 Umsetzung

Mit der Umsetzung des Konzeptes zeigt sich wie gut dieses Gelingen ist. Nachdem nun geklärt ist, wie die Anwendung aussehen soll und welche Werkzeuge verwendet werden, muss eine Testumgebung aufgebaut werden. Ebenfalls muss der Umgang mit den Frameworks und der integrated development environment (IDE), zu deutsch integrierte Entwicklungsumgebung, vertraut sein.

4.1 Aufbau einer Testumgebung

Der Server im Smarthome Lab erledigt rund um die Uhr Aufgaben. Dabei überwacht er den Zustand verschiedener Geräte und sorgt dafür dass diese erreichbar sind. Auf ihm liegen auch Medien und der Webaufttritt, welcher über einen Webserver läuft. Dies muss immer zuverlässig funktionieren. Damit der Betrieb nicht eingeschränkt ist, muss eine virtuelle Testumgebung geschaffen werden. Für diesen Zweck wurde ein Abbild des Servers erzeugt, als alles zuverlässig gearbeitet hat. Auf dem Abbild war neben dem Betriebssystem auch der Webserver mit Webaufttritt und die MySQL-Datenbank zu finden. Da diese identisch mit den Daten des Servers waren, konnte realistisch getestet werden. Durch das einfache erstellen von Sicherheitskopien konnte risikofrei Software installiert werden um zu schauen, wie sich diese auf das System auswirkt. Mit den verschiedenen Sicherheitskopien, konnte bei Problemen immer auf einen alten Stand zurückgegriffen werden.

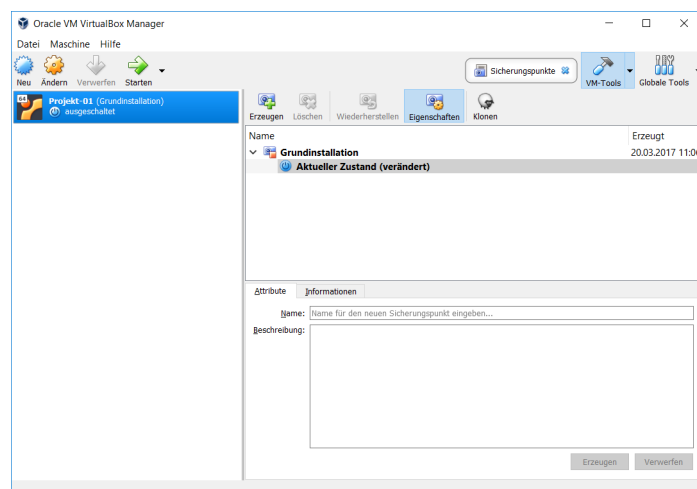


Abbildung 9: VM-Ware mit geladenem Abbild

In Abbildung 9 sieht man den VM-Ware Player. Dieser ist ein kostenloses Tool um Virtuelle Maschinen abzuspielen. Der VM-Ware Player simuliert dabei Hardware, welche im Live-betrieb verändert werden kann. [vmwareinc.2018] In diesem Fall wird der Server aus dem Smarthome Lab nur Simuliert. Dabei kann ihm mehr Speicher hinzugefügt werden, oder auch Speicher entfernt werden. So kam es zu dem Problem, dass auf der Virtuellen Maschine (VM) nicht ausreichender Festplattenspeicher zur Verfügung stand. In diesem Fall musste die komplette VM vergrößert werden. Dies ist mit VM-Ware Player nicht machbar. Unter Windows wird bei der Installation von VM-Ware Player auch der Vdiskmanager installiert. Ein Werkzeug, welches sich über die Kommandozeile steuern lässt. So lässt sich mit dem Befehl `vmware-vdiskmanager -x 100Gb vm.vmdk`, im Verzeichnis der VM, die Größe verändern. [ThomasKrennAG.2018] Damit war ein Vergrößern der VM zwar möglich, der zusätzliche Speicher stand zwar physisch, durch eine größere VM Datei zur Verfügung, konnte jedoch nicht verwendet werden. In der VM wurde ein sogenannter Snapshot verwendet. Bei einem Snapshot, handelt es sich um eine Kopie der VM zu einem bestimmten Zeitpunkt. Dieser musste ebenfalls vergrößert werden. Da die Quellen hierzu rar waren, erforderte die Lösung eine lange Suche. Die Lösung ist, mit dem Vdiskmanager den Snapshot auf die gleiche Größe zu bringen. Der Speicher wird nun angezeigt, kann jedoch noch nicht verwendet werden, da er keiner Partition zugewiesen worden ist. Dies ist über das Betriebssystem, Ubuntu Server 16.04, welches auf der VM läuft nicht einfach zu bewerkstelligen. Eine Lösung ist es Ubuntu 18.04 zu verwenden. Dabei wird ein Abbild der DVD, welche für die Installation von Ubuntu benötigt wird, in das Virtuelle DVD Laufwerk gelegt. Ubuntu kann ohne Installation verwendet werden. Es liefert das Programm GParted, mit welchem sich der neue, noch nicht zugewiesene Speicher, der Serverpartition zuweisen lässt. Mit einem Neustart der VM und dem Entfernen des Images aus dem virtuellen Laufwerk, hat die VM nun ausreichend Festplattenkapazität.[automatix.]

4.1.1 Integration in das Netzwerk

Da es sich bei der VM um ein direktes Abbild des Servers im Smarthome Lab ist, hat es auch die identischen Eigenschaften. So auch die Media-Access-Control-Adresse (MAC). Mit der MAC-Adresse soll der Netzwerkkarte eine eindeutige Identifikationsnummer zugewiesen werden. Dies geschieht schon bei der Herstellung und der Router kann damit Geräte identifizieren, welche im Netzwerk eingebunden sind. Damit ist dem Router bewusst welches Gerät, welche Pakete und welche Internet Protocoll Adresse (IP) erhält. [Zisler.2016] Solche IPs sollen weltweit eindeutig sein, sind es aber nicht, wenn es nicht gewünscht ist. Bei einem Abbild erwartet der User ein identisches System. Möchte er dieses System allerdings im gleichen Netzwerk einsetzen, so wird der Router die Pakete an Router und an die VM senden, da der Router nicht unterscheiden kann, wer der Korrekte Empfänger ist. Wird das nicht korrigiert,

kann es leicht passieren das unerwünschte Änderungen am Server und nicht an der VM unternommen werden. Denn die VM, wie auch der Server werden über Secure Shell (SSH) angesteuert. Dabei wird ein Programm verwendet, in diesem Fall Putty, welches eine sichere Verbindung über das Netzwerk zum Server oder zur VM aufbaut. Ist man nun im Netzwerk des SHL ist nicht klar ob die VM oder der Server angesprochen wird. Denn schließlich haben beide die selbe IP zugewiesen bekommen. Eine Lösung für dieses Problem wird direkt in vmware-Player zur Verfügung gestellt. In Abbildung 10 sieht man das man die Media-Access-Control-Adresse einfach in einem Textfeld anpassen kann. Mit einer neuen Media-Access-Control-Adresse unterscheidet das Netzwerk nun beide *Netzwerkkarten*. Damit erhält die Virtuelle Maschine nun eine neue IP und ist damit eindeutig adressierbar.

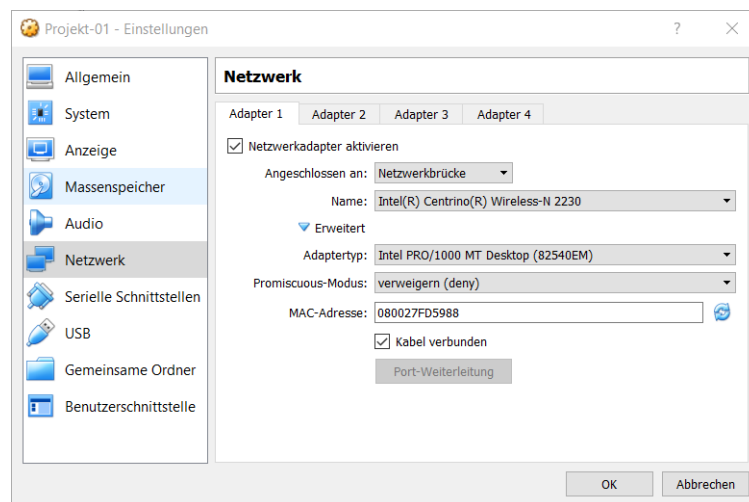


Abbildung 10: Netzwerkeigenschaften im VM-Ware-Player

4.1.2 MySQL

Auf dem Server stand bereits eine MySQL Datenbank zur Verfügung. Es handelt sich hierbei um eine relationale Datenbank. [Oracle.] MySQL steht unter der General Public License (GPL) 2.0 und ist OpenSource. WordPress setzt eine MySQL oder eine MariaDB voraus.[akbarali] Die Daten der Tabellen, welche Wordpress genutzt hatte, waren alle weiter erreichbar und konnten verwaltet werden. Die Daten für die Datenbank selbst haben gefehlt. Wenn diese Daten fehlen, so ist es möglich die Datenbank, mit Adminrechte, ohne Passwort zu starten. Nach einem solchen Start, kann das Passwort beliebig verändert werden. In diesem Fall wurde auch das von der Datenbank untersagt. Eine Datenbank war für die kommende Arbeit notwendig und somit auch der Zugriff. Da dies nicht möglich war, musste die Datenbank neu installiert werden. Für diesen Zweck wurden alle Tabellen und Datensätze in eine Datei gesichert. Über ein Plugin, welches in WordPress installiert wurde, konnte eine solche Datei erzeugen. Mit der Neuinstallation der MySQL-Datenbank wurde ein bekanntes

Passwort vergeben. Die Daten wurden importiert und es mussten nur noch wenige Daten angepasst werden, da es sich um die identischen Daten handelte.

4.1.3 Nginx

Vorinstalliert war auch der WebServer Nginx. Standardmäßig handelt es sich hier um einen Webserver, welcher HTTP anfragen entgegennimmt und GET Anfragen verarbeitet. Um einen WordPress Blog welcher PHP, JavaScript und Hypertext Markup Language (HTML) benötigt, ist das völlig ausreichen.[**akbarali**] Für weitere Verarbeitungen wie POST, PUT und DELETE müssen diverse Konfigurationen an Nginx vorgenommen werden.[**MFB.2013**]

4.2 XAMPP

Um lokal die Webanwendung entwickeln zu können, ist es nötig auf der lokalen Maschine MySQL und PHP installiert zu haben. Eine Lösung, welche für dieses Problem angeboten wird, ist XAMPP. Es handelt sich um eine frei Paketsammlung zu welcher auch PHP und MySQL gehören und wurde von den Apache Friends entwickelt.[**ApacheFriends.**] Es steht unter der GPL. Durch seine UI ist das starten von PHP oder MySQL mit einem klick erledigt. Sollte eine Datenbankabfrage mal nicht möglich sein, könnte eine Fehlerquelle sein, das der MySQL-Server nicht läuft.

4.3 Austausch von Daten

Für einen sicheren Datentransfer von der Entwicklungsmaschine zur Virtuellen Maschine wird Secure Copy (SCP) verwendet. Um dies nicht über ein Command Line Interface (CLI) machen zu müssen, gibt es Software mit einer grafischen Oberfläche. Mit solch einer Software wird eine sichere Verbindung aufgebaut. Ein Datentransfer funktioniert so ähnlich wie bei einem File Transfer Protocol (FTP) Tool. Man zieht die zu übertragenden Dateien mit der Maus in das Zielverzeichnis. Ein solcher Vertreter ist, unter dem Betriebssystem Windows, WinSCP, welches in Abbildung 11 zu sehen ist.

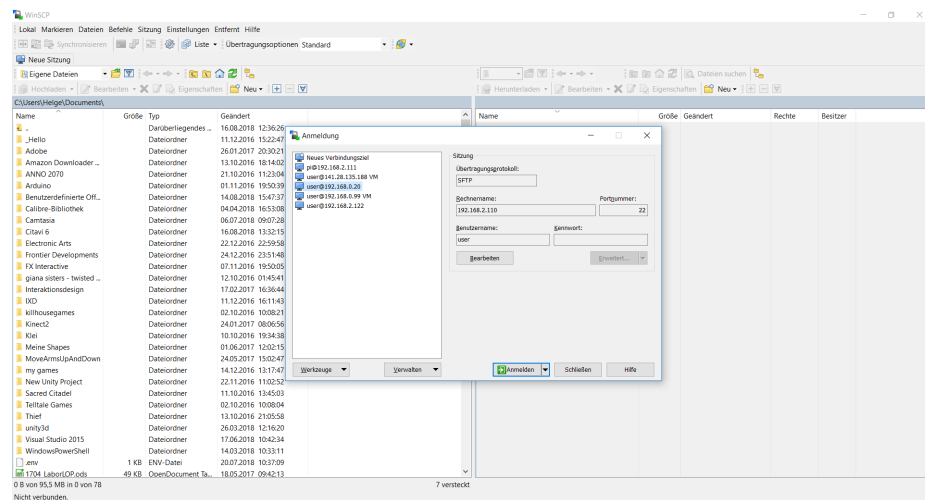


Abbildung 11: WinSCP

4.4 Frontend

Bei einem Frontend spricht man von dem was der Anwender sieht. Es handelt sich um Buttons, Links, Farben, Layouts und vieles mehr (uvm.). Wie in Kapitel 3.9 erwähnt, wurde Angular für die Umsetzung des Frontends verwendet. Zum Entwicklungszeitpunkt befand sich Angular in der Version 5, was sich im Laufe der Umsetzung geändert hat. Dies nahm jedoch keinen Einfluss auf die Entwicklung und die Webanwendung wurde mit Angular 5 fertig entwickelt. Es erfüllt die Aufgaben einer Komponentenbasierten Darstellung von verschiedenen Bausteinen. Durch seine Feste Struktur von Services, Components (zu Detusch (z. Dt.) Komponenten) und Views (z. Dt. Ansichten) [Clow.2018], ist es übersichtlich und Kenntnisse mit Angular reichen, um sich einzuarbeiten. Es wird von Google entwickelt, aber profitiert von einer sehr aktiven Community (z. Dt. Gemeinschaft). Diese entwickelt Module, welche den Funktionsumfang von Angular erweitern. Dies ermöglicht die Massachusetts Institute of Technology (MIT)-Lizenz, welche den Code frei einsehbar macht und erlaubt diesen zu verändern. [BonnyKern.2017]

4.4.1 Installation von Angular

Bevor man Angular installieren kann, muss erst einmal Node.js auf dem Entwicklungssystem installiert werden. Es ist für die gängigen Betriebssysteme wie Windows, MacOS und Linux verfügbar. Es wird kostenfrei auf <https://nodejs.org/de/> angeboten. Bei Node.js handelt es sich um eine Laufzeitumgebung. **[Node.js.2018]** Dies ist vergleichbar mit Java, welches für die Entwicklung von Programmen benötigt wird, welche in Java geschrieben sind. Im Fall von Node.js wird JavaScript beziehungsweise (bzw.) EcmaScript verwendet. Mit der Installation von Node.js wird auch der Node Package Manager (NPM) installiert. Bei NPM handelt es sich um ein Tool mit welchem Programme, Frameworks uvm. installiert werden können. Die Angebotenen Dienste und Programme werden auf <https://www.npmjs.com> angeboten. Auch Angular wird hier angeboten und damit auch unzählige Erweiterungen für Angular. Möchte man Angular nun installieren geschieht das über das Command Line Interface. Hier bei kann mittels Parameter entschieden werden ob etwas für das ganze Betriebssystem global oder nur lokal für ein einzelnes Projekt installiert werden soll. Im Falle von Angular geschieht das global. Der Befehl zum installieren findet man ebenfalls auf <https://www.npmjs.com>. Werden keine weiteren Parameter angegeben, wird immer die neuste Version heruntergeladen.

```
1 $ npm install angular -g
```

4.4.2 Commandlineinterface von Angular

Angular selbst liefert seit der Version 2 ein Command Line Interface mit. Mit diesem CLI ist es möglich Angular Projekte zu verwalten. Mit einer Zeile kann ein komplettes Projektsetup erzeugt werden. Mit Hilfe von Parameter kann angegeben werden, wie Anwendung gestylt wird. Dabei kann man beispielsweise wählen ob man die Anwendung mit Cascading Style Sheets (CSS) oder mit Syntactically Awesome Stylesheets (SASS) gestalten möchte. Bei SASS handelt es fast um die gleiche Syntax wie CSS. Es unterscheidet sich in dem Punkt, das es möglich ist Blöcke zu verschachteln und es gibt die Möglichkeit Variablen zu verwenden. Da noch kein Browser nativ SASS unterstützt, wird der SASS-Code in CSS-Code übersetzt. Angular wird über die Konsole verwendet indem ng angeführt wird. Im folgenden wird ein Beispielprojekt angelegt, welches zum Gestalten SASS verwendet. Und um nicht mit anderen HTML-Tags oder Direktiven zu kollidieren, wird jeder Anwendung für ihre eigenen Tags, was im Kapitel 4.4.5 erläutert wird, ein Prefix angelegt. Im Beispiel ist es bsp.

```
1 $ ng new Beispielprojekt --style=sass --prefix=bsp
```

Name	Änderungsdatum	Typ	Größe
.git	17.08.2018 15:18	Dateiordner	
e2e	17.08.2018 15:17	Dateiordner	
node_modules	17.08.2018 15:18	Dateiordner	
src	17.08.2018 15:17	Dateiordner	
.angular-cli.json	17.08.2018 15:17	JSON-Datei	2 KB
.editorconfig	17.08.2018 15:17	EDITORCONFIG-D...	1 KB
.gitignore	17.08.2018 15:17	Textdokument	1 KB
karma.conf.js	17.08.2018 15:17	JavaScriptdatei	1 KB
package.json	17.08.2018 15:17	JSON-Datei	2 KB
package-lock.json	17.08.2018 15:18	JSON-Datei	422 KB
protractor.conf.js	17.08.2018 15:17	JavaScriptdatei	1 KB
README.md	17.08.2018 15:17	MD-Datei	2 KB
tsconfig.json	17.08.2018 15:17	JSON-Datei	1 KB
tslint.json	17.08.2018 15:17	JSON-Datei	3 KB

Abbildung 12: Projektordner: Angularprojekt

Name	Änderungsdatum	Typ	Größe
app	17.08.2018 15:17	Dateiordner	
assets	17.08.2018 15:17	Dateiordner	
environments	17.08.2018 15:17	Dateiordner	
favicon.ico	17.08.2018 15:17	Symbol	6 KB
index.html	17.08.2018 15:17	Opera Web Docu...	1 KB
main.ts	17.08.2018 15:17	TS-Datei	1 KB
polyfills.ts	17.08.2018 15:17	TS-Datei	4 KB
styles.SASS	17.08.2018 15:17	SASS-Datei	1 KB
test.ts	17.08.2018 15:17	TS-Datei	1 KB
tsconfig.app.json	17.08.2018 15:17	JSON-Datei	1 KB
tsconfig.spec.json	17.08.2018 15:17	JSON-Datei	1 KB
typings.d.ts	17.08.2018 15:17	TS-Datei	1 KB

Abbildung 13: Sourceordner in Angular Projekt

Name	Änderungsdatum	Typ	Größe
app.component.html	17.08.2018 15:17	Opera Web Docu...	2 KB
app.component.SASS	17.08.2018 15:17	SASS-Datei	0 KB
app.component.spec.ts	17.08.2018 15:17	TS-Datei	1 KB
app.component.ts	17.08.2018 15:17	TS-Datei	1 KB
app.module.ts	17.08.2018 15:17	TS-Datei	1 KB

Abbildung 14: App Ordner in Src-Ordner des Angularprojektes

Wie man in Abbildung 12 sehen kann wird nicht nur ein Projektsetup angelegt, sondern auch ein Ordner, in dem alle NPM Abhängigkeiten verwaltet werden und es wird auch

eine Versionverwaltung mit der Hilfe von Git übernommen, sofern Git auf dem System installiert ist. Es muss nur ein Repository angelegt und mit dem Projekt verbunden werden.

4.4.3 TypeScript

Angular selbst ist TypeScript-basiert. Das bedeutet das Angular nicht direkt JavaScript verwendet, wie es noch in Angular.js gemacht worden ist, sondern es verwendet eine Art objektorientiertes JavaScript. Bei Ecma International (Ecma) handelt es sich um eine Private Organisation, welche Programmiersprachen standardisiert. So handelt es sich bei JavaScript, wie es heute geläufig ist und von den meisten Browser unterstützt wird, eigentlich um EcmaScript 5. Seit wenigen Jahren existiert EcmaScript 6. Dies ist sozusagen ein JavaScript welches Klassen, Interfaces, Arrow-Functions uvm. anbietet. Diese Features werden von vielen Entwickler gerne gesehen und lösen Probleme, mit welchen man schon länger zu tun hat und sie nur auf komplizierten Wegen lösen konnte. Allerdings unterstützen die meisten Browser noch nicht oder nur in Teilen den EcmaScript 6 Standard. An dieser Stelle kommt TypeScript zum Einsatz. Eine Installation von TypeScript läuft, wie bei Angular, über NPM. Man kann nun alle Features von EcmaScript 6 verwenden und in eine TypeScript-Datei schreiben. TypeScript übersetzt diesen Code in eine EcmaScript 5 Datei, mit welcher jeder gängige Browser umgehen kann.

Eines der größten Probleme, welche mit EcmaScript 6 gelöst wurde, ist das Problem mit *this*. Wenn ein Entwickler von einer anderen Objektorientierten Sprache zu JavaScript wechselt, kann er schnell verwirrt sein. In den meisten Programmiersprachen, entspricht das Schlüsselwort *this*, dem Objekt der Klasse, welche man gerade schreibt. So muss das nicht in JavaScript sein und kann daher schnell zu langen Fehlersuchen führen, wenn dieses Problem nicht bewusst ist. In der Programmiersprache JavaScript ist *this* in der Regel, das was vor einem Punkt einer Variablen steht. Meistens handelt es sich um das Objekt. Nun ist es aber so, das man sehr häufig mit dem Window-Objekt von JavaScript arbeitet, ohne dies zu wissen. Denn die variable *window* kann einer function, welche nicht neu entwickelt wurde, voran gestellt werden, muss es aber sehr häufig nicht. Ein bekanntest Beispiel ist die Funktion *setTimeout()*. *window.setTimeout()* ist equivalent zu *setTimeout()*. Der *setTimeout*-Funktion muss eine Funktion als Parameter mitgegeben werden. Wird hier *this* verwendet, so handelt es sich um das window-Objekt. Daher kann es passieren, das der Entwickler überzeugt ist eine Variable definiert zu haben, die Konsole ihn aber auf das Gegenteil hinweist. Dies ist ein Problem das schon sehr lange existiert und auf die verschiedensten Wege gelöst wurde. Da alte Programme, welche in JavaScript programmiert wurden, weiter Funktionieren müssen, darf mit einer neuen JavaScript Version die Funktionalität von *this* nicht geändert werden. Deshalb wurde eine neue Möglichkeit

eingeführt, mit welcher Funktionen definiert werden können. Dabei handelt es sich um die Arrow-Funktion. Der Name wurde gewählt weil sie statt dem Schlüsselwort `function` einen Pfeil verwendet. Wenn man über diese Syntax eine Funktion entwickelt und `this` verwendet, so handelt es sich um das `this` aus der Klasse und nicht um das `window`-Objekt. Im folgenden, sieht man ein Beispiel für eine solche Funktion.

```
1 () => {  
2     console.log(this.firstname)  
3 }
```

4.4.4 Module

Mit der Verwendung von Angular, ist ein Kontakt mit Modulen nicht zu Vermeiden. Bei einem Modul wird eine komplette Angularanwendung umfasst. Dazu gehören Components, Services und Direktiven. Welche in den folgenden Kapitel erläutert werden. Module können allerdings auch nur als Funktionserweiterungen entwickelt werden.[**Clow.2018**] Das Framework macht es leicht, fremde Module einzubinden. Wichtige Module werden schon von dem Framework selbst angeboten. So übernehmen diese die Netzwerkkommunikation, Formularbehandlung oder auch das Darstellen von Daten.

4.4.5 Components

Component z. Dt. Komponenten, sind wie der Name schon verrät Teilbereiche einer Angular-Anwendung. Bei Components handelt es sich, im eigentlichen Sinne, um Module. Da Components aber von sehr großer Bedeutung sind verdienen Sie ihr eigenen Abschnitt und werden in den meisten Fachbücher als eigenes Kapitel behandelt. Sie werden für die Darstellung von bestimmten Teilen Verwendet. So kann eine Component sich nur um die Ausgabe einer Tabelle zuständig sein. Dabei steht diese Component für sich alleine und ist nicht von anderen Components abhängig und beeinflusst auch nicht andere Components, wenn dieses Verhalten nicht erwünscht ist. Eine Component kann in einer fremden Component verwendet werden. Ein Verschachteln von Components ist möglich. Und auch hier stehen Sie für sich. Selbst ein gestalten in CSS ist für jede Component separat möglich. Die Ausgabe wird in HTML geschrieben und mit TypeScript-Code erweitert.[**Clow.2018**]

Das CLI von Angular unterstützt bei der Erstellung von Components den Entwickler sehr. So ist es völlig ausreichen, wenn der Benutzer beim erstellen nur die Component benennt. Angular wird für sie völlig automatisiert einen eigenen Ordner, mit CSS bzw. SASS Datei, einer HTML Datei für die Darstellung, einer Testdatei und eine TypeScript Datei an. Dabei werden die Dateien mit Beispielcode gefüllt und sind sofort

verwendbar. Angular hält sich an Namenskonventionen. So erhält jede Component das Wort Component in der Camel-Case Syntax, in welcher verschiedene Wörter in einer Variablen durch Großbuchstaben getrennt werden, als Suffix. Ist dies nicht erwünscht, lässt sich das durch Optionale angeben in der Konsole verhindern.

```
1 $ ng generate component Beispiel
```

```
1 $ ng g c Beispiel
```

In beiden Konsolenbefehle wird das Selbe ausgeführt. Die Zweite Zeile ist eine Kurzschreibweise. Das Ergebnis wird unter anderem eine Klasse mit dem Name Beispiel-Component sein.

Olge Meiering > Angular > beispielanwendung > src > app > beispiel





Name	Änderungsdatum	Typ	Größe
 beispiel.component.html	20.08.2018 10:24	Opera Web Docu...	1 KB
 beispiel.component.SASS	20.08.2018 10:24	SASS-Datei	0 KB
 beispiel.component.spec.ts	20.08.2018 10:24	TS-Datei	1 KB
 beispiel.component.ts	20.08.2018 10:24	TS-Datei	1 KB

Abbildung 15: Dateien einer Component

4.4.6 Services

Services lassen sich mit Controller aus dem Modell View Controller (MVC) Entwurfsmuster vergleichen. Ihre Aufgaben sind es Funktionen für Components oder anderen Services zur Verfügung zu stellen. In ihnen wird berechnet, analysiert oder auch verwaltet. Funktionen werden häufig genutzte Aufgaben Zentralisiert und müssen dadurch nur an einer einzigen Stelle angepasst werden, wenn es nötig wird. Die Ausgabe und die Darstellung wird in den Components Programmirt. **[Clow.2018]**

```
1 $ ng generate service Beispiel
```

```
1 $ ng g s Beispiel
```

Auch bei den Services wird beim Erstellen alles nötige generiert und die Camel-Case-Syntax kommt zum Einsatz. In diesem Fall würde der Service BeispielService heißen.

!lge Meiering > Angular > beispielanwendung > src > app

Name	Änderungsdatum	Typ	Größe
beispiel	20.08.2018 10:24	Dateiordner	
app.component.html	17.08.2018 15:17	Opera Web Docu...	2 KB
app.component.SASS	17.08.2018 15:17	SASS-Datei	0 KB
app.component.spec.ts	17.08.2018 15:17	TS-Datei	1 KB
app.component.ts	17.08.2018 15:17	TS-Datei	1 KB
app.module.ts	20.08.2018 10:24	TS-Datei	1 KB
beispiel.service.spec.ts	20.08.2018 10:27	TS-Datei	1 KB
beispiel.service.ts	20.08.2018 10:27	TS-Datei	1 KB

Abbildung 16: Dateien eines Services

4.4.7 Direktiven

Direktiven können eine Ansicht verändern oder mehr Informationen weitergeben. Es wurden keine Direktiven in dieser Arbeit selbst entwickelt, jedoch sind sie ein wichtiger Teil, da viele Fremddirektiven zum Einsatz kamen. Direktiven werden in zwei Gruppen geteilt. Zum einen haben wir die Attributsdirektiven. Dazu gehören auch die selbst Geschriebenen Components. Sie heißen Attributsdirektiven, weil Sie wie ein Attribut einem HTML-Tag mitgegeben werden. Ein Beispiel hierfür ist das `ngClass`. Es kann durch den Programmcode die Klasse eines Tags verändern. Eine sehr häufig genutzte Direktive ist die Strukturdirektive. Sie verändert die Struktur des HTML-Codes. Strukturdirektiven erkennt man an dem vorangestellten `*`. So ist es möglich mit `*ngFor` ganze Arrays direkt im HTML-Code auszugeben oder mit `*ngIf` HTML-Code zu entfernen und nicht nur auszublenden. [Clow.2018]

```
1 $ ng generate directive Beispiel
```

```
1 $ ng g d Beispiel
```

Direktiven werden ebenfalls auf den fast gleichen Weg erstellt. Und der Name wäre in diesem Fall `DirectivBeispiel`. So können alle zusammengehörenden Bereiche den gleichen Namen haben und doch eine andere Aufgabe übernehmen.

ge Meiring > Angular > beispielanwendung > src > app

Name	Änderungsdatum	Typ	Größe
beispiel	20.08.2018 10:24	Dateiordner	
app.component.html	17.08.2018 15:17	Opera Web Docu...	2 KB
app.component.SASS	17.08.2018 15:17	SASS-Datei	0 KB
app.component.spec.ts	17.08.2018 15:17	TS-Datei	1 KB
app.component.ts	17.08.2018 15:17	TS-Datei	1 KB
app.module.ts	20.08.2018 10:29	TS-Datei	1 KB
beispiel.directive.spec.ts	20.08.2018 10:29	TS-Datei	1 KB
beispiel.directive.ts	20.08.2018 10:29	TS-Datei	1 KB
beispiel.service.spec.ts	20.08.2018 10:27	TS-Datei	1 KB
beispiel.service.ts	20.08.2018 10:27	TS-Datei	1 KB

Abbildung 17: Dateien einer Direktiven

4.4.8 Routing

Das Routing übernimmt die Aufgabe die Uniform Resource Locator (URL) einfach darzustellen. Bei Angular handelt es sich um eine Single-Page-Application. Auch wenn hierfür nur eine einzige HTML Seite benötigt wird, soll der User nicht alle auf nur einer Seite sehen. Die Anwendung Verteilt sich in Unterseiten. Der Inhalt wird nur durch JavaScript ausgetauscht. Um eine bestimmte seite Aufrufen zu können und sich dabei nicht durch viele Menüs klicken zu müssen, kann das Routing eingesetzt werden. Angular weiß anhand der URL welcher Teil angezeigt werden soll.[Clow.2018] Ein Beispiel für eine solche URL ist *www.angularbeispielseite.de/unterseite*.

4.5 Backend

Das Inventarsierungssystem kommt mit einem Fronent alleine nicht aus. Jedes größere Projekt teilt sich auch in ein Backend auf. Berechnungen, der Login und auch Netzwerkanfragen sind Dinge, die im Hintergrund der Anwendung geschehen. Der User soll von diesen dingen nichts mitbekommen. Für ihn müssen sie, wie selbstverständlich Funktionieren. In Kapitel 3.9 wurde die Große vielfalt der Frameworks erwähnt. Es wurde auch kurz erläutert warum welches Framework gewählt wurde. In diesem Kapitel soll detaillierter in das Framework Laravel und seine Backendaufgaben geblickt werden.

4.5.1 Laravel

Mit PHP existiert eine sehr mächtige Programmiersprache mit unzähligen Funktionen. Seit einigen Jahren ist PHP objektorientierter. Dadurch fanden viele Entwickler

schneller einen Zugang zu diesem mächtigen Werkzeug. Durch den riesen Umfang den PHP nativ anbietet, leidet auch die Übersichtlichkeit. HTML und PHP kann sehr unübersichtlich vermischt werden, Datenbankabfragen sind häufig lange und kompliziert und das Entwickeln von einem LogIn-System kann viel Zeit in anspruch nehmen. Genau hier setzt das PHP Framework Laravel ein. Mit seiner Renderengine Blade wird der vermischte Code übersichtlicher, die Datenbankabfragen werden kürzer und einfacher und ein LogIn-System ist schnell erstellt. Zusätzlich nutzt Laravel das bekannte MVC Muster.[**Laravel.2018**]

4.5.2 Installation von Laravel

Wie für JavaScript, gibt es auch für PHP einen Packagemanager. Dieser lädt Abhängigkeiten welche für Laravel nötig sind. Wie auch bei Angular, können hier Abhängigkeiten nachinstalliert werden. Durch die große Community sind auch viele Erweiterungen frei zur Verfügung. Im ersten Schritt muss allerdings Composer installiert werden. Die Software steht kostenfrei auf <http://www.composer.com> zur Verfügung. Diese wird für Windows, Linux und MacOS angeboten. [**Composer.2018**] Unter Windows läuft die Installation über einen Assistenten. Ist eine PHP Version welche neuer als 7.1.2 ist und die Extensions: OpenSSL, PHP: Data Objects (PDO), Mbstring Tokenizer, Extensible Markup Language (XML), Ctype und JavaScript Object Notation (JSON) installiert, kann ein Laravel Projekt mit der Versionsnummer 5.6 mit Composer angelegt werden. Zuerst muss Laravel, wie Angular erst installiert werden. Das geschieht mit folgender Zeile.

```
1 $ composer global require "laravel/installer"
```

Ein Anlegen eines Projektes kann nun von Laravel übernommen werden oder von Composer, was die folgenden zwei Beispiele demonstrieren.

```
1 $ laravel new beispiel
```

```
1 $ composer create-project --prefer-dist laravel/laravel beispiel
```

Dabei werden alle nötigen Pakete heruntergeladen und installiert.

```
1 Loading composer repositories with package information
2 Installing dependencies (including require-dev) from lock file
3 Package operations: 70 installs, 0 updates, 0 removals
4 - Installing doctrine/inflector (v1.3.0): Loading from cache
5 - Installing doctrine/lexer (v1.0.1): Loading from cache
6 - Installing dragonmantank/cron-expression (v2.2.0): Loading from
  cache
7 - Installing erusev/parsedown (1.7.1): Loading from cache
```

```
8 - Installing vlucas/phpdotenv (v2.5.1): Downloading (100%)
9 - Installing symfony/css-selector (v4.1.3): Downloading (100%)
10 - Installing tijsverkoyen/css-to-inline-styles (2.2.1): Loading
    from cache
11 - Installing symfony/polyfill-php72 (v1.9.0): Downloading (100%)
12 - Installing symfony/polyfill-mbstring (v1.9.0): Downloading
    (100%)
13 - Installing symfony/var-dumper (v4.1.3): Downloading (100%)
14 - Installing symfony/routing (v4.1.3): Downloading (100%)
15 - Installing symfony/process (v4.1.3): Downloading (100%)
16 - Installing symfony/polyfill-ctype (v1.9.0): Downloading (100%)
17 - Installing symfony/http-foundation (v4.1.3): Downloading (100%)
18 - Installing symfony/event-dispatcher (v4.1.3): Downloading
    (100%)
19 - Installing psr/log (1.0.2): Loading from cache
20 - Installing symfony/debug (v4.1.3): Downloading (100%)
21 - Installing symfony/http-kernel (v4.1.3): Downloading (100%)
22 - Installing symfony/finder (v4.1.3): Downloading (100%)
23 - Installing symfony/console (v4.1.3): Downloading (100%)
24 - Installing egulias/email-validator (2.1.5): Downloading (100%)
25 - Installing swiftmailer/swiftmailer (v6.1.2): Downloading (100%)
```

Hierbei handelt es sich um eine Auswahl der installierten Komponenten. Es handelt sich bei der Standardinstallation um 70 Komponenten. Dazu gehören Polyfills, welche Lösungen für ältere Browser, welche nicht neuen Code unterstützen, bieten, aber auch das Routing und Mailfunktionen werden übernommen.

Name	Änderungsdatum	Typ	Größe
app	21.08.2018 09:12	Dateiordner	
bootstrap	21.08.2018 09:12	Dateiordner	
config	21.08.2018 09:12	Dateiordner	
database	21.08.2018 09:12	Dateiordner	
public	21.08.2018 09:12	Dateiordner	
resources	21.08.2018 09:12	Dateiordner	
routes	21.08.2018 09:12	Dateiordner	
storage	21.08.2018 09:12	Dateiordner	
tests	21.08.2018 09:12	Dateiordner	
vendor	21.08.2018 09:13	Dateiordner	
.editorconfig	21.08.2018 09:12	EDITORCONFIG-D...	1 KB
.env	21.08.2018 09:13	ENV-Datei	1 KB
.env.example	21.08.2018 09:12	EXAMPLE-Datei	1 KB
.gitattributes	21.08.2018 09:12	Textdokument	1 KB
.gitignore	21.08.2018 09:12	Textdokument	1 KB
artisan	21.08.2018 09:12	Datei	2 KB
composer.json	21.08.2018 09:12	JSON-Datei	2 KB
composer.lock	21.08.2018 09:12	LOCK-Datei	141 KB
package.json	21.08.2018 09:12	JSON-Datei	1 KB
phpunit.xml	21.08.2018 09:12	XML-Dokument	2 KB
server.php	21.08.2018 09:12	JetBrains PhpStorm	1 KB
webpack.mix.js	21.08.2018 09:12	JavaScriptdatei	1 KB
yarn-error.log	21.08.2018 09:12	Textdokument	2 KB

Abbildung 18: Root Ordner einer Laravelanwendung

In Abbildung 18 sieht man das Wurzelverzeichnis, welches mit dem Befehl *laravel new beispiel* angelegt wurde. Man kann diesem Projektsetup ein sehr ausführliche Struktur entnehmen. Fremdbibliotheken werden streng getrennt und auch Dateien welche wirklich nach *außen* geleitet werden, liegen in einem separatem Verzeichnis. Der Datei *package.json* kann man entnehmen, das auch für eine Laravelanwendung, Erweiterungen über NPM installiert werden können. Laravel bietet eine Schnittstelle für Frontends, wobei Laravel in dieser Arbeit nur ein Backend dient. Sehr wichtig für den Entwickler ist die Datei *.env*. Der Punkt am Anfang des Dateinamen ist für ein Betriebssystem wie Windows irrelevant. Wichtig ist er in Betriebssystemen wie MacOS oder Linux. Hier werden solche Dateien *versteckt*. Diese sollen nicht versehentlich verändert werden, da sie sehr sensible Daten enthalten. Der Dateiname steht für environment, z. Dt. Umgebung. Die komplette Laramgebung nimmt sich diese Datei als Grundlage. Es werden Konfigurationen für Pfade, Datenbanken und Mailserver eingetragen. Diese stehen im Klartext dar. [Laravel.2018] Wenn diese nun auf einem Gitserver landen, könnte das folgen von Angriffen und ausspähen von Daten haben. Aber wie auch in Angular, wird in der *.gitignore* Datei verhindert, wie in Kapitel 2.1 erläutert wurde.

4.5.3 Artisan

Mit der Installation von Laravel, wird Artisan mitgeliefert. Es handelt sich über ein eigenständiges Konsolenwerkzeug wie es auch in Angular vorhanden ist. Damit lassen sich Controller und Models erzeugen. Auch das Erzeugen von Views ist möglich, aber in der Arbeit nicht nötig. Mit Artisan kann auch ein lokaler Server gestartet werden. Auch übernimmt Artisan das Migrieren von Datenbanken, welche in Laravel erzeugt worden sind. Artisan gibt die Möglichkeit Mails zu versenden und Scheduler (z. Dt. Planer) zu starten. Dabei kann ein bestimmter Ablauf geplant werden. So kann eine Aufgabe immer zu einer bestimmten Zeit, zu einem bestimmten Datum oder in einem bestimmten Intervall ausgeführt werden. Sollte der Entwickler Test geschrieben haben, können diese nicht nur mit Artisan angestoßen sondern auch analysiert und überwacht werden. Auch wenn eine lange Liste an nützlicher Befehle vorhanden sind, so ist es möglich, dass diese nicht ausreichen. Hierfür gibt es eine Schnittstelle, mit welcher man Artisan neue Befehle beibringen kann.[Laravel.2018] Artisan wird dabei mit PHP gestartet. Im folgen sieht man den Befehl um einen Controller zu erstellen.

```
1 $ php artisan make:controller BeispielController
```

Anders als Angular, übernimmt Laravel keine Unterstützung bei der Namenskonvention. An die muss der Entwickler sich selbst halten.

4.5.4 Controller

Des öfteren wurde schon das MVC Muster in dieser Arbeit erwähnt. Ich möchte nun näher auf das C eingehen. Im klassischen MVC Muster kommuniziert der Controller nur mit dem Model und verarbeitet dessen Daten. [Goll.2014] Im Grunde ist es bei dieser Anwendung genau das Selbe. Zusätzlich liefert der Controller seine verarbeiteten Informationen an die Representational-State-Transfer (REST)-Schnittstelle. Im folgenden sieht man einen leeren Controller, wie er von Artisan erstellt wurde. Folgt man dem Pfad *PROJEKT/App/Http/Controller*, so findet man alle Controller.

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class BeispielController extends Controller
8 {
9     //
10 }
```

4.5.5 REST

Representational-State-Transfer gilt als eines der bedeutendsten Leitfäden bei der Entwicklung von Verteilten Anwendungen. [Nguyen.2017] Das Inventarverwaltungssystem ist keine Verteile Anwendung. Jedoch setzt REST den Schritt das es in Zukunft möglich ist. Mit REST wird auf einem Server ein Service zur Verfügung gestellt, welcher über eine URL aufgerufen wird. So wird ein Befehl ausgeführt und berechnet. Die Ressourcen auf dem Endgerät sind wenig bis gar nicht belastet. Somit können auf Leistungsschwachen Endgeräten komplizierte Berechnungen durchgeführt werden und es werden nur die Ressourcen des Servers belastet. Die Ergebnisse können auf die Verschiedensten Art und Weisen nach außen getragen werden. Die bekanntesten Vertreter sind XML, einfacher Text und JSON. Im Fall des Inventarisierungssystem wurde JSON gewählt. Angular kann nativ mit JSON umgehen und es schnell verarbeiten. So spart man sich ein zusätzliches Modul. Für REST werden mehrere Funktionen bereitgestellt Daten an den Server zu senden und Daten vom Server zu empfangen. Die am gebräuchlichsten Funktionen sind:

- GET: zum reinem Empfangen von Daten.
- POST: zum Senden von Daten. Es können aber auch Ergebnisse zurück gesendet werden.
- PUT: zum Aktualisieren von Daten. Es können aber auch Ergebnisse zurück gesendet werden.
- DELETE: zum Löschen von Daten. Es können aber auch Ergebnisse bzw. Rückmeldungen zurück gesendet werden.

Erstellt man nun folgende Methode in einem Controller,

```
1 public function helloWorld(){  
2  
3     return response()->json(['message' => 'Hello World'],200);  
4  
5 }
```

kann in der Datei *api.php* eine URL angegeben werden über welche die Methode aufgerufen wird. Der Code zeigt wie eine JSON erstellt wird, welche als Wertepaar nur message und Hello World hat. Der Code wird nur ausgeführt wenn der Statuscode 200 ist.

```
1 Route::get('/hello', [  
2     'uses' => 'BeispielController@helloWorld'  
3 ]);
```

Diesem Stück Code, lässt sich entnehmen das die Funktion über die URL mit der Endung `/hello` aufgerufen wird. Vor dem `@` in `uses` wird der Controller bekannt gegeben und hinter dem `@` wird gesagt welche Methode zum Einsatz kommen soll.



Abbildung 19: Ergebnis von einer REST-URL

In Abbildung 19 ist zu sehen, das man über die URL `http://127.0.0.1:8000/api/hello` eine JSON erhält mit welcher man nativ weiterarbeiten kann.

4.6 Sicherheit

Auch wenn es sich hierbei um nicht sensible Daten hält, ist eine Sicherheit der Daten nicht unwichtig. Dabei unterstützt der Browser, aber vor allem das Backend erfüllt die Aufgaben der Datensicherheit. Aber auch der Betreiber der Anwendung ist durch die Datenschutz-Grundverordnung dazu verpflichtet. [intersoftconsulting.2018]

4.6.1 CORS

Cross-Origin Resource Sharing (CORS) ist eine Möglichkeit für Browser und Anwendungen interaktiv zu kommunizieren. [Cors.2018] Warum das nötig ist, wird schnell klar wenn man sich mit der Sicherheit beschäftigt. Schädlicher Code kann mittel JavaScript so eingeschoben werden das der User das nicht mitbekommt. Deswegen greift hier Same-Origin-Policy (SOP) ein. SOP verbietet eine solche Kommunikation. [Mozilla.2018] Eine Möglichkeit doch zu kommunizieren bietet CORS dabei muss genau angegeben wer Kommunizieren darf und was kommuniziert werden darf. Das geschieht in Header. Hier wird definiert um was für eine Kommunikation es sich handelt. Ein Beispiel sind Authentifizierungen oder das Versenden von JSON Daten. Die Kommunikation kann sich auf alle Anwender beschränken, es ist aber auch möglich Kommunikationen nur von einer fest definierten IP zuzulassen.

4.6.2 Token

Die Möglichkeiten sich unbefugt in ein Konto einzuwählen sind heutzutage scheinbar grenzenlos. Hier werden die Entwickler stark gefordert sichere Lösungen zu konzipieren und umzusetzen, da die Daten der User wichtig und sensibel sind. Sie müssen geschützt werden, denn der User vertraut darauf. Ein Schutz ist ein Token. Der

Token ist ein sehr langer String. Dieser wird angelegt, sobald sich der User angemeldet hat. Dieser Token wird in einer Datei temporär auf dem Endgerät des Users abgelegt. Aber auch der Server und somit die Anwendung kennen diesen Token. [StiftungWarentest.2015] Warum ist ein Token nun nötig und warum reichen die Login Daten nicht aus? Ist der User eingeloggt, kann eine Fremde Person sich als diese ausgeben und benötigt nicht das Passwort. Was sie allerdings benötigt ist den Token. Dieser ist sehr kryptisch und lange. Ein Programm diesen String erraten zu lassen ist fast unmöglich und würde bei den vielen verschiedenen Kombinationen, mit heutiger Technik, Jahrzehnte benötigen. Ein weiterer Vorteil ist es, dass der User die Seite schließen kann und bei einem neuen Öffnen wieder in seiner persönlichen Umgebung befindet. Der Token wird nach einer definierten Zeit ungültig und es muss ein neuer Token generiert werden. Auch mit einem Logout wird der Token gelöscht und ungültig.

4.6.3 Salt

Registriert sich ein User in einer Anwendung oder auf einer Website, so werden seine Login Daten gespeichert. Damit diese nicht im Klartext einzusehen sind, werden diese gehasht. Bei dem sogenannten Hashing werden Wörter oder Passwörter in eine kryptische Zeichenkette übersetzt. In der Regel wird niemand etwas mit der Zeichenkette anfangen können. Sollte jemand unbefugt Zugriff auf die Datenbank zugreifen können oder sollte der Datenbankadministrator diese Daten ungefragt einsehen wollen, so kann festgestellt werden, welche User das selbe Passwort verwenden. Denn der Hash wird bei dem gleichen Wort, auch den gleichen Hashwert haben. Die gängigste Lösung ist der sogenannte Salt (z. Dt. Salz). Dabei bekommt der Hashwert eine zufällig generierte Zeichenkette vorangestellt. Damit wird der Hashwert ein weiteres mal verändert. [intersoftconsultingservicesAG.2013] Identische Passwörter ähneln sich in keiner Weise.

	id	personid	email	firstname	lastname	password	role	annotation	remember_token	created_at	updated_at
8	852852		test@test.de	Jonny	Test	\$2y\$10\$9#kTpR5KvWmW5MP1VCAOQnX0qZK72eMf...	User	(NULL)	(NULL)	2018-07-16 10:17:49	2018-07-16 10:17:49
9	963		neu@google.de	Neu	Test	\$2y\$10\$DwqfMvFOAhuv8PxCqYruqY7Zha8PM6UT3m...	Admin	(NULL)	(NULL)	2018-07-16 10:55:51	2018-07-16 10:55:51
10	587895		mai@lab.de	Mai	Phong	\$2y\$10\$K7.Td0.JDyHyq28X1W3IOuZv4U0W56hXy6Hn...	User	(NULL)	(NULL)	2018-07-22 14:55:31	2018-07-22 14:55:31
11	321		31@test.com	Hanns	Manns	\$2y\$10\$IQ./mppm.f2n2j2HmuBuTek0ldZ5PKZdT8mkQDS...	User	(NULL)	(NULL)	2018-07-23 15:58:09	2018-07-23 15:58:09

Abbildung 20: Alle User nutzen das Passwort "test"

4.7 Deploy

Deploy ist das englische Wort für bereitstellen. Und nichts anderes wird bei einem Deploy gemacht. Die entwickelte Anwendung wird auf dem Server bereitgestellt. Dabei sind bei der Arbeit mit Frameworks Dinge zu beachten, auf diese kurz eingegangen werden soll.

4.7.1 Deploy in Angular

Ein Deploy in Angular ist ohne viel Aufwand verbunden. Mit dem Befehl *ng build* wird ein dist Ordner erzeugt. Angular wandelt hierbei die TypeScript Dateien in JavaScript Dateien um. In diesem ist eine HTML Datei und eine Optimierte JavaScript Datei. Diese Lädt man auf dem Webserver und die Anwendung läuft.

4.7.2 Deploy in Laravel

Auch wenn der Webserver PHP Unterstützt ist ein einfaches Hochladen der Anwendung nicht die Lösung. Zwar muss sie nicht erst, wie bei Angular, die Anwendung gebaut werden, aber der Server muss so konfiguriert werden, das mit REST-Anfragen umgegangen werden kann. Bei der Konfiguration konnte für den Laraverteil trotz ausführlicher Anleitungen kein Erfolg erzielt werden. Im Grunde Funktioniert die Laravelanwendung, kann aber keine PUT, POST und DELETE Befehle entgegen nehmen. **[Stauffer.2017]** GET Befehle sind möglich und Methoden können ausgeführt werden. So kann die Datenbank manuell gefüllt werden. Dies ist leider nicht die Lösung für das Problem. Um die Anwendung zum laufen zu bringen ist es nötig das ein Experte sich näher mit dem Deploy beschäftigt. Auch wenn die Daten über ein FTP Programm hochgeladen werden, ist es einfacher die Anwendung über Git zu installieren.

5 Anwendung

Dieses Kapitel soll die Interessantesten Teile der Anwendung Dokumentieren. Mit dem Vorangegangenen Code lassen sich alle nicht dokumentierte Teile im Code sehr gut nachvollziehen.

5.1 Anlegen, Aktualisieren, Löschen und Ausgeben von Gegenständen

Im Frontend werden neue Gegenstände über ein Formular gesendet. Die Inhalte des Formulars werden per JSON mittels POST an die Restschnittstelle von Laravel gesendet, welche der Controller verarbeitet und in die Datenbank einträgt. Sollte ein Barcode schon vorhanden sein, wird der User darauf hingewiesen. Bei einer Löschung, Aktualisierung oder einer Ausgabe muss nur darauf hingewiesen werden, um welches Item es sich genau handelt.

```
1 public function postItem(Request $request)
2 {
3     $item = new Item();
4     $item->barcode = $request->input('barcode');
5     $item->name = $request->input('name');
6     $item->description = $request->input('description');
7     $item->type = $request->input('type');
8     $item->room = $request->input('room');
9     $item->status = $request->input('status');
10    $item->annotation = $request->input('annotation');
11    $item->image = $request->input('image');
12    $item->lend = $request->input('lend');
13    $item->manufactor = $request->input('manufactor');
14    $item->save();
15    return response()->json($item, 201);
16
17 }
18
19 public function getItems(){
20     $items = Item::all();
21     $response = [
```

```

22 'items' => $items
23 ];
24 return response()->json($response,200);
25 }
26
27 public function putItem(Request $request, $id){
28 $item = Item::find($id);
29 if(!$item){
30 return response()->json(['message'=>'Document not found Barcode:
    '. $id . '. The number is not the Barcode. Look into the
    Table'], 400);
31
32 }
33 $item->name = $request->input('name');
34 $item->description = $request->input('description');
35 $item->type = $request->input('type');
36 $item->room = $request->input('room');
37 $item->status = $request->input('status');
38 $item->annotation = $request->input('annotation');
39 $item->image = $request->input('image');
40 $item->lend = $request->input('lend');
41 $item->manufactor = $request->input('manufactor');
42 $item->save();
43 return response()->json(['item' => $item], 200);
44 }
45
46 public function putItemLendBack( $id){
47 $item = Item::find($id);
48 if(!$item){
49 return response()->json(['message'=>'Document not found
    putItemLendBack: '. $id . '. The number is not the Barcode.
    Look into the Table'], 400);
50
51 }
52
53 $item->status = 'back';
54
55 $item->save();
56 return response()->json(['item' => $item], 200);

```

```
57 }
58
59 public function deleteItem($id){
60     $item = Item::find($id);
61     $item->delete();
62     return response()->json(['message' => 'Deleted'],200);
63
64 }
65
66 public function getItem($id){
67     $item = Item::find($id);
68     if(!$item){
69         return response()->json(['message'=>'Document not found ID: '.
            $id . '. The number is not the Barcode. Look into the Table'
            ], 400);
70
71 }
72
73 $response = [
74     'items' => $item
75 ];
76
77 return response()->json($response,200);
78
79
80 }
```

New Item

Barcode *

Item Name *

description

Type ▼

Room ▼

Status ▼

annotation

manufactor

Datei auswählen Keine ausgewählt

Save Item

Abbildung 21: Anlegen von Items

5.2 Ausgabe von allen Gegenständen

Um eine Liste aller Gegenstände wird vom Server eine JSON generiert, welche mit in Angular in einer Tabelle ausgegeben werden. Hierfür wurde Angular Material verwendet, welches die Tabelle auch gleichzeitig gestaltet hat.

```

1 <mat-table [dataSource]='dataSource' matSort>
2
3 <ng-container matColumnDef="barcode">
4 <mat-header-cell *matHeaderCellDef mat-sort-header>Barcode</mat-
  header-cell>
5 <mat-cell *matCellDef="let items">{{items.barcode}}</mat-cell>
6 </ng-container>
7
8 .
9 .

```

```

10 .
11
12 <mat-header-row *matHeaderRowDef="displayedColumns"></mat-header-
    row>
13 <mat-row *matRowDef="let row; columns: displayedColumns;"></mat-
    row>
14
15 </mat-table>

```

5.3 Routing Strategie

Wie in Kapitel 3.1 beschrieben, wird der Webaufttritt von WordPress verwaltet. WordPress selbst nutzt das Routing um Verschiedene Seiten anzeigen zu können. Läuft eine Angular Anwendung auf dem gleichen Server, oder soll sie gar in WordPress eingebunden werden, kann das zu Problemen führen, da sie die gleiche Strategie nutzen. Eine Unterseite in Angular aufzurufen ist nicht möglich wenn man es zuvor nicht konfiguriert. WordPress wird immer eine Fehlerseite werfen, sofern nicht eine Unterseite von WordPress den gleichen Namen trägt. Aber auch in diesem Fall wird eine Falsche Seite dem User Präsentiert.

Eine Lösung ist die *HashLocationStrategy*. Bei dieser Strategie muss der Anwendung mit einem Konfigurationsbefehl mitgeteilt werden, wie das Routing ab sofort funktionieren soll. Dabei wird vor der dem eigentlich Anhängen der Unterseite ein Rautensymbol angehängen. Damit kann Wordpress in der Regel nur die Startseite Anzeigen und wird nicht auf eine Fehlerhafte Seite verweisen. Somit ist ein Routing unter der gleichen Domain möglich und es ist trotz allem für den User nachvollziehbar.

```

1 RouterModule.forRoot(routes, {useHash: true})

```

Es ist nur useHash im Routermodul auf true zusetzen und Angular erledigt den rest. Die variable routes ist ein Array, welches alle Routen verwaltet.

5.4 E-Mail Reminder

Für den Reminder (z. Dt. Erinnerer) wird eine Konfiguration angelegt, in welcher definiert wird was in der E-Mail stehen soll, welche der User erhalten wird. Dabei handelt es sich um ein Template. Alle nötigen Daten holt sich Laravel selbständig aus der Datenbank. Ein Anlegen eines solchen Schedulers wird im folgenden Code Beispiel Demonstriert.

```

1 protected function schedule(Schedule $schedule)
2 {

```

```
3         $schedule->command('reminder:mail')->daily();
4     }
```

5.5 Datenbank import und export

Für einen unerfahrenen User soll es leicht sein Daten zu importieren und für eine Sicherung oder zur Nutzung ohne Netzwerkverbindung zu nutzen. Auch kann es sein, das viele Daten Angelegt werden sollen, welche einfacher oder schneller über eine Tabelle angelegt werden könnten. Dabei soll der User keine Daten anlegen oder mit Programmen arbeiten müssen, welche ihm unbekannt sind. Eine sehr Bekannte Möglichkeit solche Dateien zu erstellen ist Microsoft Excel. Der User soll eine Excel-Datei erstellen und diese über ein grafische Benutzeroberfläche importieren können. Dabei darf es keine Rolle spielen welche Endung die Excel-Datei hat. Mit einem Export soll ebenfalls eine Excel-Datei erzeugt werden.

Für den Import wurde mittels Composer die Erweiterung PHPOffice verwendet. Es kann Excel-Dateien lesen und schreiben. Mit dem Lesen werden die Daten in die Datenbank geschrieben. Hierbei wird Dopplung vermieden und wenn es sich um bekannte deutsche Begriffe handelt, werden diese auf englisch in die Datenbank geschrieben. Sollte ein Begriff unbekannt sein, so wird der deutsche Wortlaut übernommen. Um Dopplung zu vermeiden musste ein eindeutiger Schlüssel gewählt werden. Die Entscheidung fiel auf den Barcode. Hat ein neu angelegtes Item keinen Barcode, wird es nicht importiert. Dieser ist zwingend nötig. Der Export wird ebenfalls von PHPOffice übernommen. Hierbei muss nur Angegeben werden, wie jede Spalte heißt. Der User kann nun sich einmalig eine Tabelle Exportieren lassen und diese nach diesem Schema füllen und Importieren. Die Exportieren Funktion findet der User Unter dem Button Backup.

```
1 public function insertItemsInDatabase()
2 {
3     $inputFileName = public_path() . '\table\barcode.xlsx';
4     $inputFileType = IOFactory::identify($inputFileName);
5
6     $reader = new \PhpOffice\PhpSpreadsheet\Reader\Xlsx();
7
8
9     $reader->setReadFilter(new MyReadFilter());
10
11     //Einlesen
12     $spreadsheet = $reader->load($inputFileName);
13 }
```



```

14      $sheetData = $spreadsheet->getActiveSheet()->toArray(null,
15          true, true, true);
16
17      //alle datein eintragen und übersetzen
18      foreach ($sheetData as $row){
19
20          $name = "";
21          $barcode = "";
22          $description = "";
23          $room = "";
24          $status = "";
25          $type = "";
26          $annotation = "";
27          $image = "";
28          $lend = "";
29          $manufactor = "";
30
31
32
33          //Übersetzung
34          if($row['F'] == 'Zubehör'){
35              $type = 'equipment';
36          }
37
38          ...
39
40          //Abbruchkreterium
41          if (DB::table('items')->where('barcode', '=',
42              $barcode)->count() > 0) {
43              continue;
44          }
45          else{
46              DB::table('items')->insert(
47                  ['barcode'=> $barcode, 'name' => $name, '
48                      description' => $description,
49                      'type' => $type, 'room' => $room, 'status'
50                      => $status]

```

```

49         );
50     }
51
52
53     }
54
55     return response()->json($sheetData, 200);
56
57 }

```

```

1  public function exportItemsInXml(){
2  //Überschriften für die Spalten
3  $spreadsheet = new Spreadsheet();
4  $sheet = $spreadsheet->getActiveSheet();
5  $sheet->setCellValue('A1', 'Artikel');
6  $sheet->setCellValue('B1', 'Barcode');
7  $sheet->setCellValue('C1', 'Bezeichnung');
8  $sheet->setCellValue('C1', 'Bezeichnung');
9  $sheet->setCellValue('D1', 'Einsatzbereich');
10 $sheet->setCellValue('E1', 'Position');
11 $sheet->setCellValue('F1', 'Kategorie');
12 $sheet->setCellValue('G1', 'Status');
13
14 $items = Item::all();
15
16 $exportFileName = "table\\barcode_export.xlsx";
17
18 //Eintragen der Daten
19 foreach ($items as $item){
20 $sheet->setCellValue('A' . $index, $item->name);
21 $sheet->setCellValue('B' . $index, $item->barcode);
22 $sheet->setCellValue('C' . $index, $item->description);
23
24 $sheet->setCellValue('D' . $index, $item->room);
25 $sheet->setCellValue('E' . $index, 'unknown');
26 $sheet->setCellValue('F' . $index, $item->type);
27 $sheet->setCellValue('G' . $index, $item->status);
28 $index++;
29 }

```

```

30
31 $writer = new Xlsx($spreadsheet);
32
33
34
35 $writer->save($exportFileName);
36
37
38
39 //in eine Datei schreiben.
40 return response()->download(public_path('table\\barcode_export.
    xlsx'))->deleteFileAfterSend();
41
42 }

```

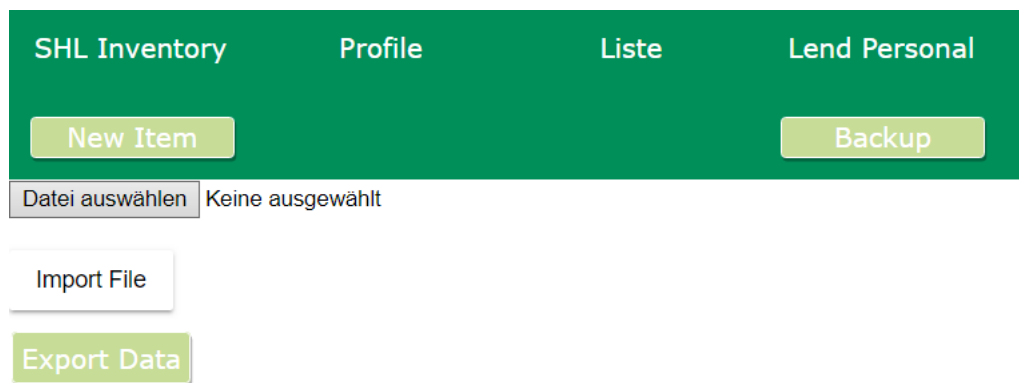


Abbildung 22: Importieren und Exportieren von Items

5.6 Gestalten der Navigationsleiste

Die Webanwendung ist nicht responsiv. Was heißt das sich nicht auf allen Endgeräten perfekt dargestellt und nur auf einem PC oder Tablett komfortabel genutzt werden kann. Jedoch wurde mittels CSS3 eine Navigation gestaltet, welche sich der Größe des Browsers anpasst.

```

1 .felx-container {
2     display: flex;
3     background-color: #018F59;
4     width: 100%;
5     margin-left: auto;
6     margin-right: auto;

```

```
7      flex-direction: row;
8      flex-wrap: wrap;
9      justify-content: space-between;
10
11     span {
12         color: white;
13         margin: .8rem;
14         width: 120px;
15         text-align: center;
16
17
18     }
19
20 }
```

6 Diskussion der Ergebnisse

Die Arbeit verlief zu großen Teilen sehr gut. Auf einem lokalen System läuft Sie reibungslos und erfüllt ihre Aufgaben. Dies ist auf dem Server nicht der Fall. Das Deploy ist gescheitert und somit auch die Implementierung. Sollte ein Deploy noch nachträglich umgesetzt werden, so handelt es sich hierbei um ein funktionierende Inventursystem für das Smarthomelab der Hochschule Furtwangen.

6.1 Fazit

Für mich ist das Projekt ein Erfolg. Es hat mich in meinem Wissen weitergebracht und ich habe eine Anwendung umgesetzt, welche alle Phasen durchlaufen ist. Schade ist es, dass es am Ende an dem Deploy gescheitert ist. Aber aus diesen Fehlern habe ich gelernt. Ich hätte mit einem Zwischenergebnis die Tücken eines Deploys feststellen können und wäre darauf besser vorbereitet. Auch wenn die Konzeptionszeit sehr gut verlief, würde ich diese in einem zweiten Anlauf verkürzen. Nichts desto weniger bin ich sehr zufrieden mit dem Ergebnis.

6.2 Ausblick

Auf dieser Arbeit lässt sich aufbauen. So wurde eine REST-Schnittstelle entwickelt. Hier könnte man Ansetzen und einen Barcodescanner entwickeln welcher auf diese Schnittstelle zugreift. Dabei spielt es noch nicht einmal eine Rolle um welche Programmiersprache es geht, solange sie REST-Befehle senden kann. Auch könnte mit einer Schnittstelle zu OpenHAB die Status der einzelnen Items erweitert werden. So dass man in der Tabelle schon sehen kann welches Item in Betrieb ist. Als Beispiel könnte man zeigen, welche Lampen in welcher Farbe leuchten.

Eidesstattliche Erklärung

Ich versichere, dass ich die vorstehende Arbeit selbständig verfasst und hierzu keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Stellen der Arbeit die wörtlich oder sinngemäß aus fremden Quellen entnommen wurden, sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt oder an anderer Stelle veröffentlicht.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben kann.

Winden im Elztal, den 23. August 2018 Helge Meiering

A Anhang

CD mit dem Gesamten Code und der LaTeX-Dateien