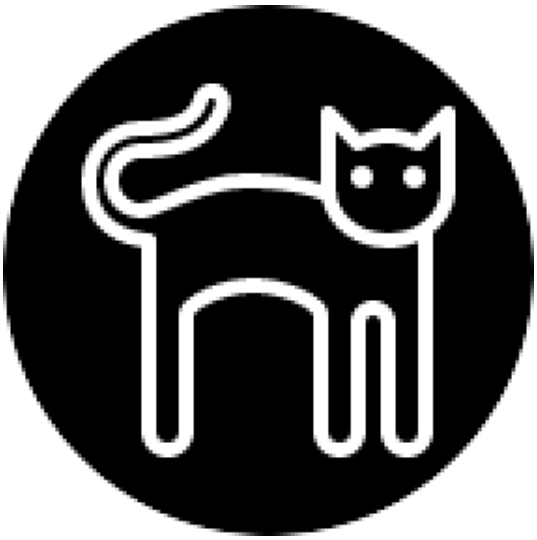

Katzenfütterungsanlage

HTBLA Kaindorf an der Sulm
Grazer Straße 202, A-8430 Kaindorf an der Sulm
Ausbildungsschwerpunkt Mechatronik und Automatisierungstechnik

Florian Greistorfer, Florian Harrer, Dominik Pichler, Julian Wolf

Abgabedatum: 05.04.2018

Betreut von:
Dipl.-Ing. Manfred Steiner
Dipl.Ing. Dr. Gerhard Pretterhofer
Otto

Inhaltsverzeichnis

KAPITEL 1

Webserver und Client

1.1 Begriffserklärungen

1.1.1 Server

Ein Computer oder Programm, der oder das Zugriff auf eine Resource oder einen Dienst in einem Netzwerk ermöglicht

1.1.2 Client

Ein Computer oder Programm, der oder das auf einen Server Zugreift

1.2 Anforderungen

1.2.1 Webserver

Auf der Katzenfütterungsanlage läuft ein Webserver, der es ermöglicht, dass der Benutzer das Gerät über das Internet erreichen kann. Hauptaufgaben des Servers sind dabei, Daten bereitzustellen, zu verabeiten und zu speichern und den Webclient zur Verfügung zu stellen.

1.2.2 Client

Der Client soll dem Benutzer ermöglichen, die Katzenfütterungsanlage über einen Webbrowser zu steuern. Ein Benutzername und ein Passwort sind erforderlich, damit man das Gerät bedienen kann. Das Design soll eindeutig und übersichtlich gehalten sein. Auf der Startseite sollen die eingestellten Fütterungszeiten zu sehen sein und eine allgemeine Übersicht. Über eine Navigationsleiste sollen die weiteren Seiten erreichbar sein:

- Fütterrungszeiten
- Positionsinfo
- Geräteinfo
- Update

1.3 Voruntersuchung

1.3.1 Typescript

Typescript ist eine Weiterentwicklung der Sprache Javascript, die strenge Datentypen hat. Typescript muss von einem Transpiler (=Übersetzer) in Javascript übersetzt werden. Javascript kann direkt von jedem herkömmlichen Browser ausgeführt werden.

1.3.2 Node.js

Node.js ist eine Laufzeitumgebung, die es ermöglicht, dass Javascript direkt auf einem Rechner ausgeführt werden kann.

1.3.3 Angular 2/4

Angular ist ein Typescript Framework, das aus dem Javascript-Framework AngularJS weiterentwickelt wurde. Es wird von Google entwickelt. Angular ist gegliedert in Komponenten. Die grobe Struktur wird in der Abbildung 1.1 dargestellt.

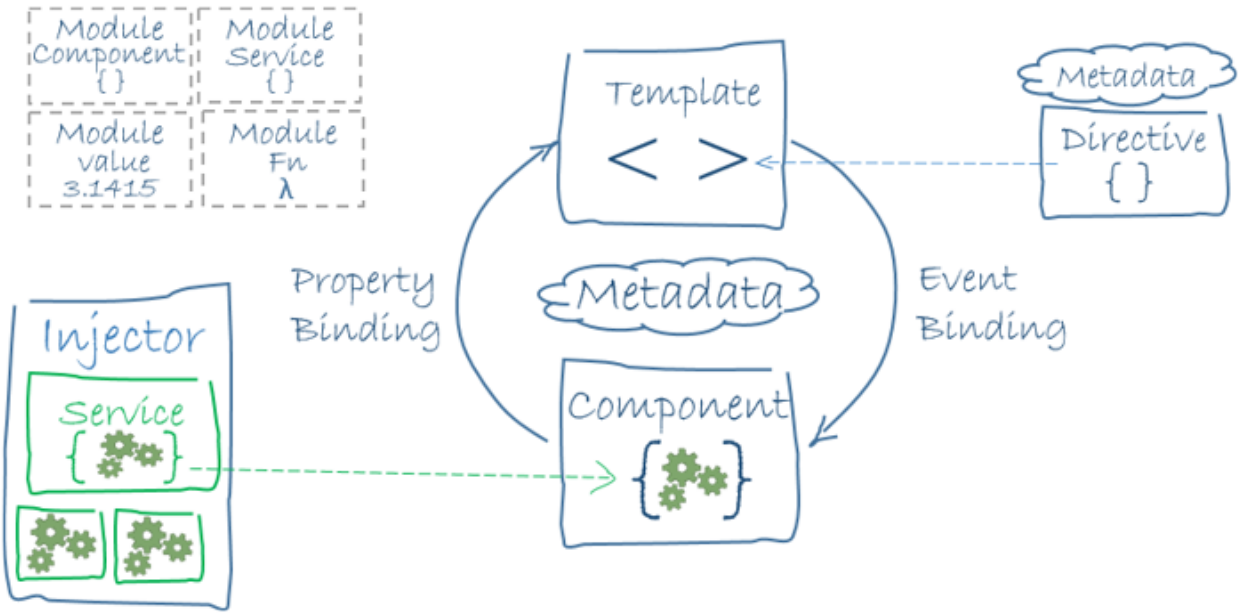

Abbildung 1.1: Angular Struktur

1.3.3.1 Modules

Jede Angular App ist in *Module*s gegliedert. *Module*s fassen meist ähnliche Funktionen zusammen. Jede App muss mindestens ein *Module* enthalten. Dies heißt standartmäßig *AppModule*. Ein *Module* ist die größte Einheit einer Angular App. Ein *Module* kann folgende Komponenten beinhalten:

- Components
- Services

Angular selbst besteht ebenfalls aus *Module*s, den sogenannten Libraries. Der Name jedes *Module*s beginnt mit *@angular* z.B. *@angular/core*.

1.3.3.2 Components

Ein *Component* kontrolliert einen Teil des Bildschirms, den sogenannten *view*. Die Logik des *Component*s wird in einer Klasse definiert. Die Klasse interagiert mit dem *view* durch eine Application Programming Interface (API) von Eigenschaften und Methoden.

1.3.3.3 Templates

Das Aussehen des *view*s wird in einem *Template* definiert. Ein *Template* ist eine HyperText Markup Language (HTML) Datei, mit Angular's Template Syntax. Das bedeutet, dass einige Zusatzbefehle vorkommen können. Beispiele hierfür sind:

***ngFor**
Stellt ein HTML Element so oft dar, wie Elemente in einem Array vorhanden sind.

Listing 1.1: *ngFor Beispiel

```
1 <a *ngFor="let lang of languages" href="{{lang.href}}">{{lang.text}}</a>
```

***ngIf**

{{variable}}

(click)

[variable]

<app-route>

1.3.3.4 Data binding

1.3.3.5 Services

1.3.4 Bootstrap

Bootstrap

1.4 Umsetzung

1.4.1 Client

1.4.1.1 Design

Das Design sollte übersichtlich und einfach gestaltet werden. Der Benutzer soll auf den ersten Blick die wichtigsten Funktionen und Informationen erkennen können.

Auf der Starseite sind alle wichtigen Informationen übersichtlich dargestellt. Auf der linken Seite werden die Uhrzeit der letzten erfolgreichen Fütterung, die Zeit der nächsten Fütterung und die Zeit bis zur nächsten Fütterung dargestellt. Darunter werden Fehler und Warnungen, fals welche auftreten sollten, angezeigt. Da unbekannt ist, wie viele Fehler und Warnungen auftreten, werden diese in einem *ngFor aufgelistet. Auf der rechten Seite sind die aktiven Fütterungszeiten aufgelistet.

Abbildung 1.2: Startseite

Auf der Fütterungszeiten-Seite kann der Benutzer die Katzenfütterungsanlage ein und ausschalten. Dies wurde mit einer Checkbox realisiert, die durch Styles wie ein Schalter gestaltet wurde. Darunter können die Fütterungszeiten geändert und deaktiviert werden. Siehe Abbildung 1.3. Der Button 'Speichern' wird deaktiviert, sobald eine Zeit ungültig eingegeben wurde, oder wenn die Zeiten nicht in aufsteigender Reihenfolge sortiert sind.

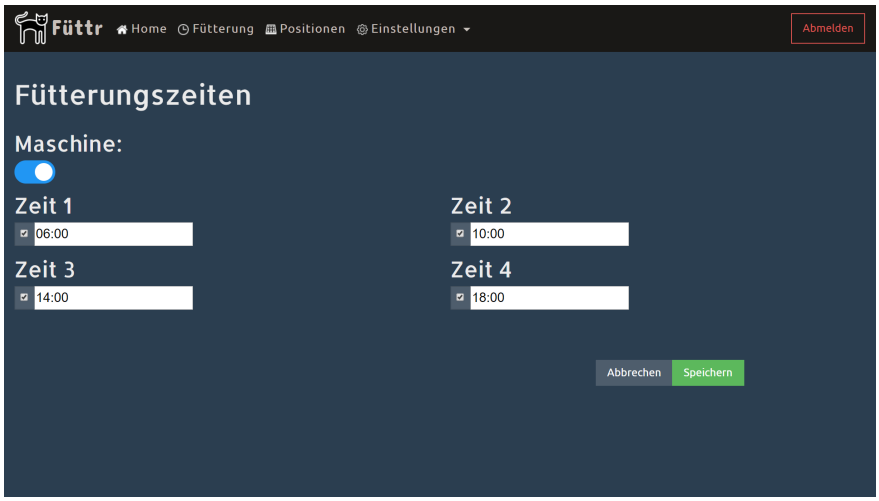

Abbildung 1.3: Fütterungszeiten

Auf der Geräteinformations-Seite werden die wichtigsten Daten über das Gerät ange-

zeigt. diese Daten sind:

- Seriennummer
- Interner Rechner
- WLAN Status
- IP Adresse
- Softwareversion

Die Seriennummer ist eindeutig und wird von einem Server zugeteilt. Die IP-Adresse ist die aktuelle **externe** IP-Adresse.

Auf der Update-Seite kann der Benutzer nach Updates suchen, Updates starten oder die Maschine herunterfahren. Wenn der Benutzer auf den Herunterfahren-Button klickt, der sich auf der linken Seite ganz unten befindet, wird er gewarnt, dass die Maschine nur mehr über das Aus- und wieder Einstecken des Netzteils startbar ist.

Abbildung 1.5: Update

1.4.1.2 Funktion

1.4.2 Server

1.4.2.1 Funktion

1.4.2.2 Mongodb

1.4.2.3 Kommunikation mit dem Java Programm

1.5 Zusammenfassung und Verbesserungsmöglichkeiten

Anhang

ANHANG A

Abbildungsverzeichnis

ANHANG B

Tabellenverzeichnis

ANHANG C

Listings

ANHANG D

Abkürzungsverzeichnis

API Application Programming Interface 3

HTML HyperText Markup Language 3