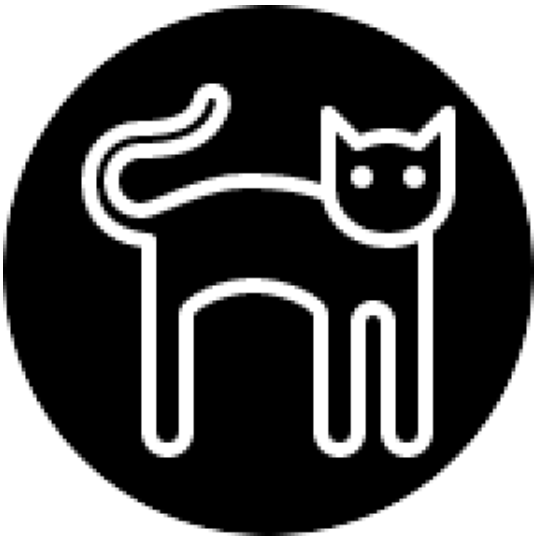

Katzenfütterungsanlage

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.

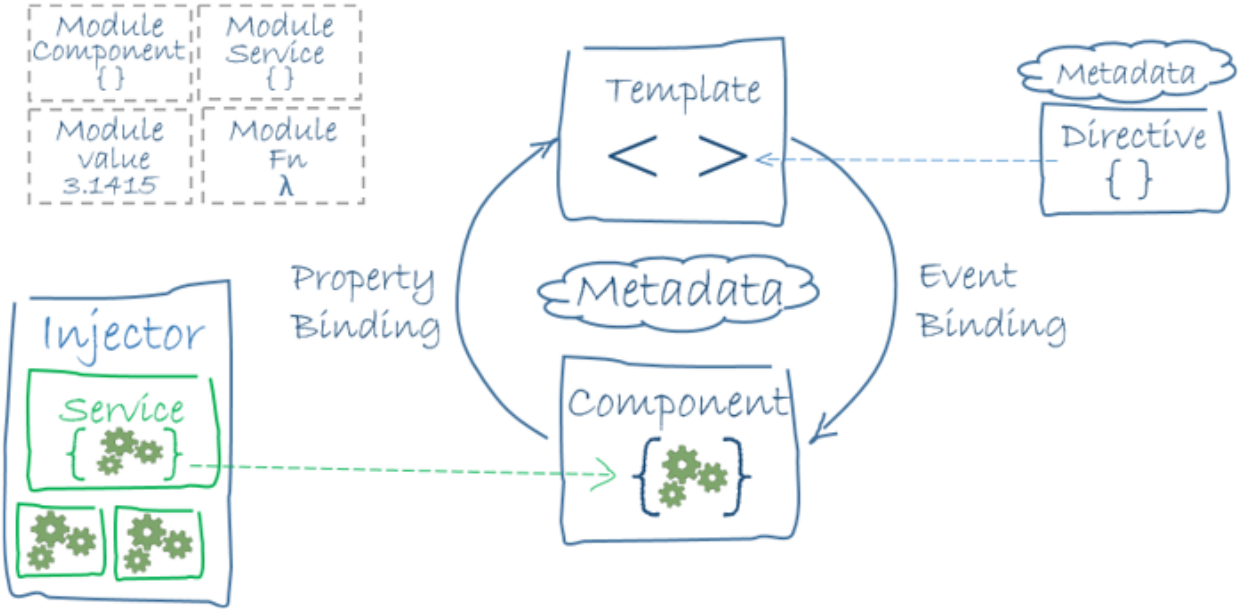

Abbildung 1.1: Angular Struktur

1.3.3.1 Modules

1.3.3.2 Components

1.3.3.3 Templates

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.

Abbildung 1.2: Startseite

Auf der Starseite sind alle wichtigen Informationen übersichtlich dargestellt. Auf der linken Seite werden ... Auf der rechten Seite sind die aktiven Fütterungszeiten aufgelistet. Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Listing 1.1: Errors and Warnings in einem *Ngfor

```
1  <ul style="list-style-type: none; margin-left: -40px;">
2    <li *ngFor="let warning_message of ErrorsAndWarnings.Warnings">
3        <div class="alert alert-warning alert-dismissable fade show" [hidden]="
             warning_message.hidden">
4          <button type="button" class="close">
5              <span aria-hidden="true" (click)="ack(warning_message)">&times;<
                   /span>
6            </button>
7            <strong i18n>Warning! </strong>{{warning_message.message}}
8        </div>
9    </li>
10   <li *ngFor="let error_message of ErrorsAndWarnings.Errors">
11       <div class="alert alert-danger alert-dismissable fade show" [hidden]="
             error_message.hidden">
12         <button type="button" class="close">
```

```
13              <span aria-hidden="true" (click)="ack(error_message)">&times;</s
                    pan>
14            </button>
15            <strong i18n>Error! </strong>{{error_message.message}}
16          </div>
17      </li>
18  </ul>
```

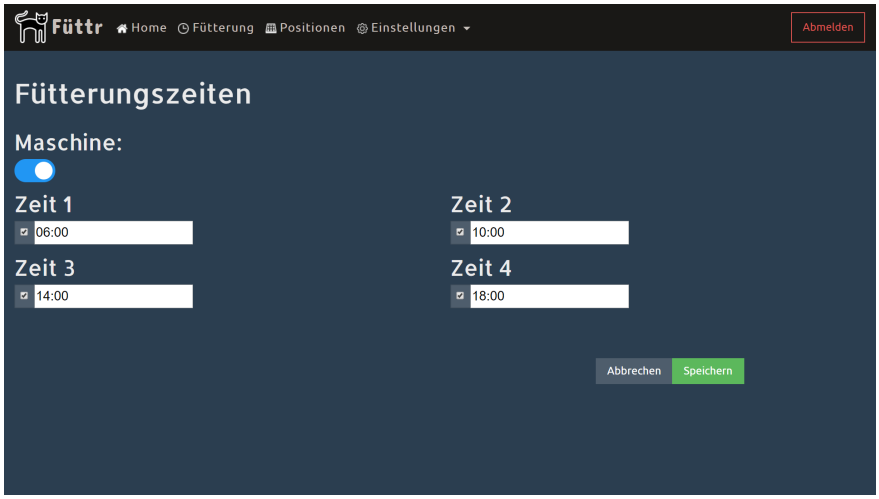

Abbildung 1.3: Fütterungszeiten

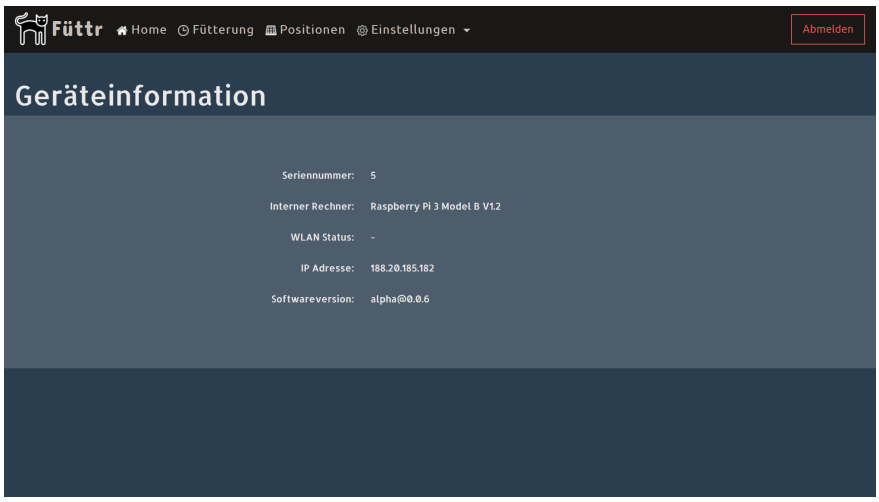

Abbildung 1.4: Geräteinformationen

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 Verbesserungs-möglichkeiten

Anhang

ANHANG A

Abbildungsverzeichnis

ANHANG B

Tabellenverzeichnis

ANHANG C

Listings