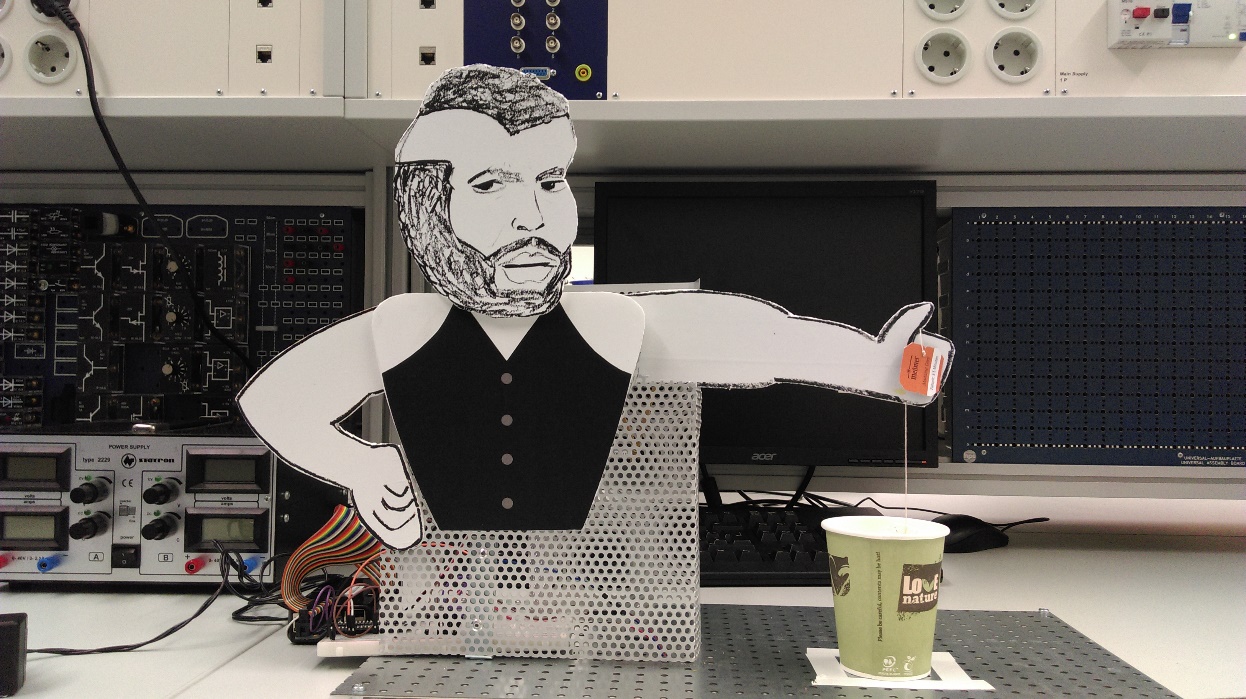
**Technische Dokumentation**

**Mr. Tea**



Artur Tahiraj 2009542 | Stefanie Kohl 2011097 | Oliver Völling 2248958 | Irena Becker 2238833

1. Technische Umsetzung der Hardware mit einem Raspberry Pi
2. Programmierung und Gestaltung der App

Die App ermöglicht zunächst die Bedienung des Roboterarms. Außerdem wird der Nutzer über die App über den aktuellen Stand der Teezubereitung informiert und erhält eine entsprechende Mitteilung, wenn der Tee auf die trinkbereite Temperatur abgekühlt ist.

Die Interaktion über ein Smartphone bietet sich an dieser Stelle an, da im aktuellen digitalen Zeitalter in jedem Haushalt mindestens ein Smartphone vorhanden ist. Die Kommunikation über das Smartphone bietet dem Nutzer die Möglichkeit, sich weiteren Verpflichtungen zu widmen und übernimmt die Aufgabe der Teezubereitung.

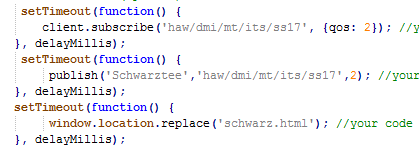
Um die Applikation so plattformunabhängig zu implementieren, entschieden wir uns für eine Umsetzung in HTML und JavaScript.

Im ersten Schritt programmierten wir eine Webanwendung. Als die Funktionalität und die Verbindung zum Server/Broker implementiert waren, gestalteten wir das Frontend nach unseren Vorstellungen weiter aus. Zuletzt wandelten wir unsere Webapplikation mit Hilfe von *Cordova* und *PhoneGap* in eine native App für Android Smartphones um. Durch unsere Webanwendung haben nun auch Nutzer von iPhones und WindowPhones die Möglichkeit unseren Roboter zu bedienen.

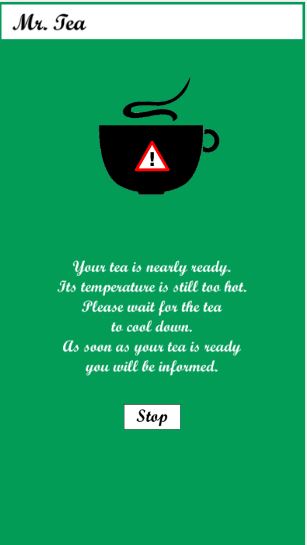
Beim Start der App wird noch keine Verbindung zum Broker aufgebaut. Erst wenn eine Teesorte ausgewählt wird, verbindet sich die App mit dem Broker, submittet den zuvor definierten Raum und sendet die Information über die gewählte Teesorte an den Broker.

Da die Schritte 1) subscriben, 2) publishen und 3) der Wechsel auf den neuen Screen zeitlich genau in dieser Reihenfolge erfolgen sollen, haben wir einen Delay von jeweils 100 Millisekunden zwischen die einzelnen Vorgänge integriert.



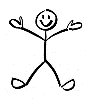


Im weiteren Verlauf „wartet“ die App auf die Information „fertig gezogen“ im Broker und ändert in diesem Fall den Screen. Der Nutzer wird darüber informiert, dass der Tee zwar fertig, jedoch zu heiß und noch nicht trinkbereit ist. Wenn das Thermometer die vordefinierte Temperatur an den RaspberryPi ünergibt, wird die Information „Tee trinkbereit“ in den Broker gegeben. Die Anwendung liest die Information aus und gibt sie an den Nutzer weiter. Die Anwendung kann zu jeder Zeit abgebrochen werden. Dabei wird die Last-Will Nachricht "abbr“ an den Broker gesendet und die Verbindung zum Broker wird zurückgesetzt.



***Abbildungen:*** *Benutzerscreens mit 1) Wahl des Grünen Tees, 2) Abwarten der Ziehzeit, 3) Abwarten der Abkühlung auf trinkbare Temperatur, 4) Information über die Fertigstellung des Tees*

Ein Sequenzdiagramm veranschaulicht die zeitliche Abfolge der Vorgänge folgendermaßen:



Motor

Broker

Raspberry

Thermometer

Motor runter

Grüner Tee

Grüner Tee

Countdown

Motor unten

Motor hoch

Fertig gezogen

Fertig gezogen

Motor oben

Temperatur

Temperatur

Temperatur

Tee trinkbereit

Tee trinkbereit

Folgende Zustände ergeben sich hieraus:

Countdown fertig

Ziehen

Heiß

Abkühlen

Abbruch

Teesorte selektiert

Abbruch

Fertig

Reset

Start

Zusammenfassend können wir festhalten, dass die Umsetzung der App mit HTML und JavaScript eine übersichtliche Aufteilung in Frontend und Logik ermöglichte. Auch unser erstmaliger Einsatz von Cordova und PhoneGap erwies sich als nutzerfreundlich und unkompliziert. Um die Ladezeiten der Screens zu verringern hätten wir von Anfang an unsere App nach dem Prinzip einer Single-Page Anwendung implementieren können. Außerdem werden wir in Zukunft Push-Nachrichten anzeigen lassen, statt aufwändige Screenwechsel für gewünschte Brokerinhalte zu programmieren.