

Projektkonzept

Tee Maschine - Mr. Tea

1) Name

Von Beginn an war uns wichtig die geplante Anwendung so ansprechend und unterhaltsam wie möglich umzusetzen. Nachdem wir uns über die Grundfunktion geeinigt hatten, entschieden wir uns für die Findung eines passenden Namens. Ein teaminternes Brainstorming ergab folgende Nennungen: Teetunker, Teebeutelunkmaschine, The Tea Machine, Teeinator, Py-Tea, Tea-Man, Mr. Tea.

Letzterer erschien uns als optimaler Arbeitsnamen für unser zukünftiges Projekt.

2) Projektziel

Mr. Tea soll zunächst die Aufgabe erfüllen den optimalen Tee aufzubereiten. Er soll dem Nutzer die Beachtung der Ziehzeit des jeweiligen Tees abnehmen und somit verhindern, dass der Tee misslingt. Die detaillierten Anforderungen an Mr. Tea sind in unserer nachstehenden Anforderungsanalyse beschrieben.

3) Anforderungsanalyse

Wir stellen uns vor, dass sich der Nutzer unserer Anwendung für einen Teebeutel seiner favorisierten Teesorte entscheidet (zunächst werden wir von zwei vordefinierten Teesorten ausgehen) und diesen am Tunk-Arm von Mr. Tea befestigt. Außerdem schenkt er heißes Wasser in eine Tasse ein und stellt diese auf die vorgesehene Stelle unterhalb des Tunk-Arms.

Ab hier soll der Nutzer ausschließlich per App mit Mr. Tea kommunizieren. Er gibt an, für welche Teesorte er sich entschieden hat und alles weitere übernimmt Mr. Tea.



Abbildung 1: Startbildschirm der App



Abbildung 2: Auswahl der Teesorte

Zunächst wird die Temperatur des Wassers via Infrarot Thermometer gemessen. Liegt diese in einem gewissen Rahmen, taucht Mr. Tea den Teebeutel am Tunk-Arm mit Hilfe eines Motors in das heiße Wasser. Ist das Wasser zu kalt, wird der Nutzer darüber informiert und

aufgefordert das Wasser erneut zu erhitzen. Ist das Wasser noch zu warm, wartet Mr. Tea die Abkühlung des Wassers ab.

Ist der Teebeutel in das heiße Wasser eingetaucht, wartet Mr. Tea die jeweilige Ziehzeit ab und entnimmt den Teebeutel anschließend. Zeitgleich erhält der Nutzer eine Push-Nachricht per App, dass sein Tee nun verzehrbereit ist.



Abb. 3: Warten auf optimale Wassertemperatur



Abb.4: Countdown der verbleibenden Ziehzeit

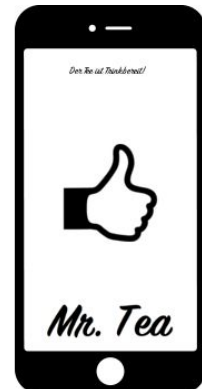


Abb. 5: Tee ist fertig

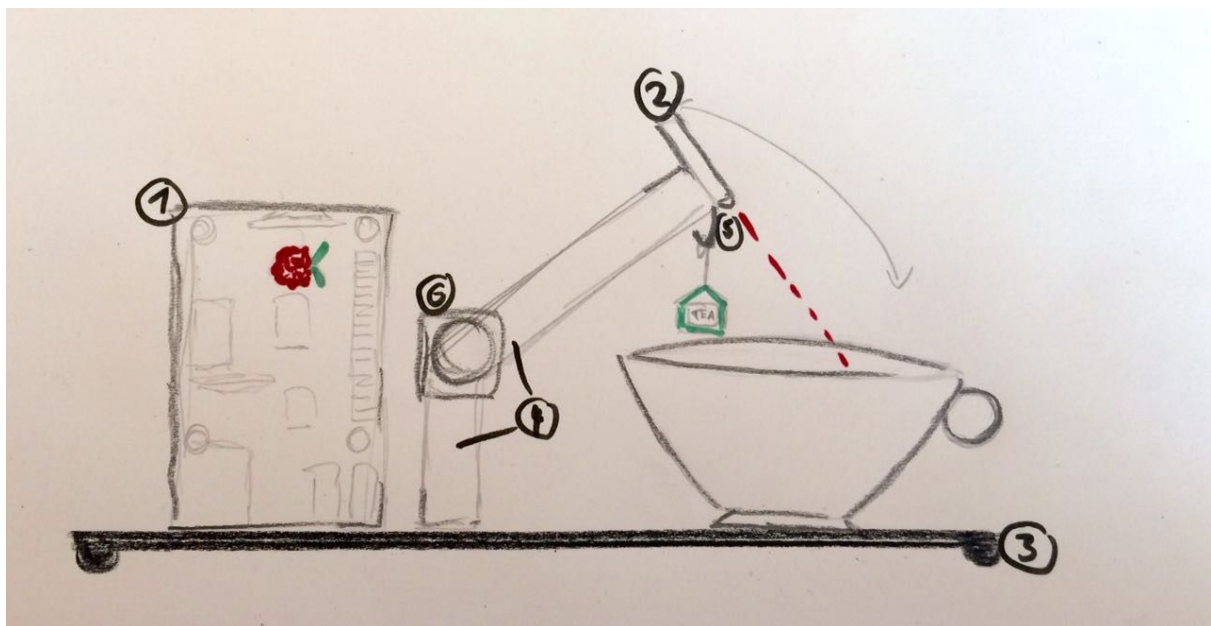


Abbildung 6: Schematische Umsetzung des Mr. Tea

1: Raspberry Pi, 2: Pyrometer, 3: Platte, 4: Tunk - Arm, 5: Halterung Teebeutel, 6: Motor

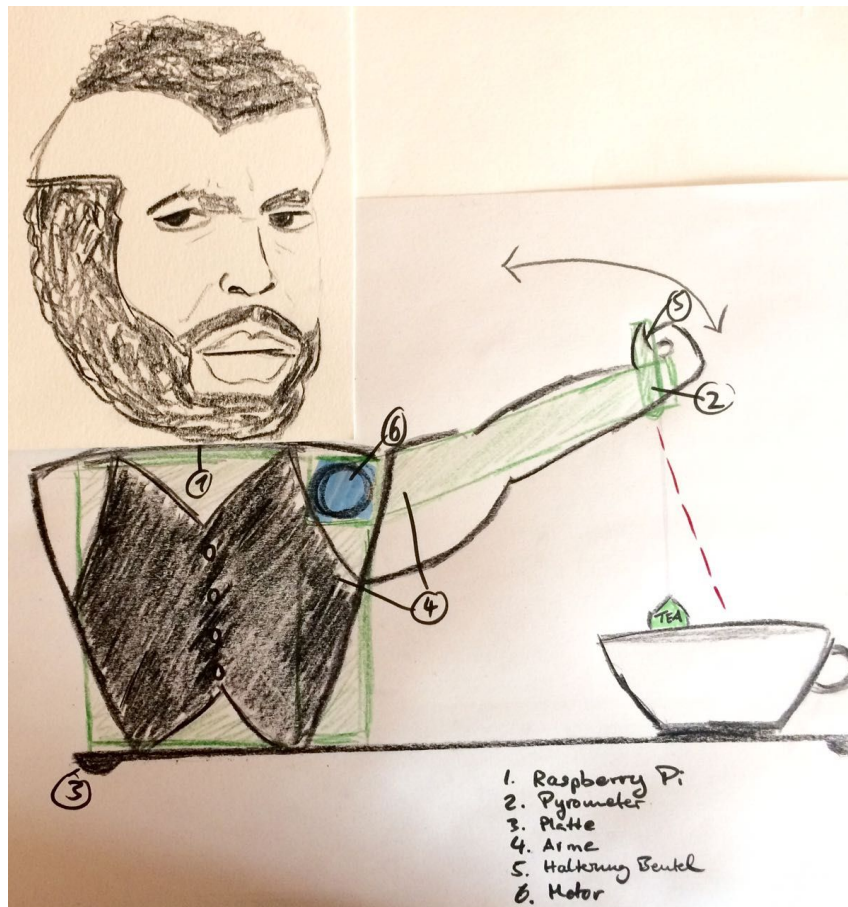


Abbildung 7: Skizze einer eventuellen Umsetzung

Um die Anforderungen zu veranschaulichen haben wir einige User Stories verfasst:

- Als Nutzer möchte ich bei der Zubereitung meines optimalen Tees unterstützt werden.
- Als Nutzer möchte ich meinen jeweiligen Tee nur mit der optimalen Wassertemperatur zubereiten.
- Als Nutzer möchte ich die Ziehzeit meines Tees nicht immer im Auge behalten müssen.
- Als Nutzer möchte ich bei Bedarf die verbleibende Ziehzeit auf meinem Smartphone angezeigt bekommen.
- Als Nutzer möchte ich über die Fertigstellung des Tees über mein Smartphone benachrichtigt werden, damit ich währenddessen anderen Tätigkeiten nachgehen kann.
- Als Nutzer möchte ich eine Teesorte meines Beliebens auswählen können.

4) Mögliche Zusatzfunktionen

Falls wir die Grundfunktion des Mr. Tea zeitnah umsetzen, behalten wir uns vor eventuelle Zusatzfunktionen zu implementieren:

- Integrierter Wasserkocher
- Automatische Teesortenselektion (Nutzer muss den Teebeutel nicht manuell am Arm befestigen)
- Einstellung der Teemenge in ml

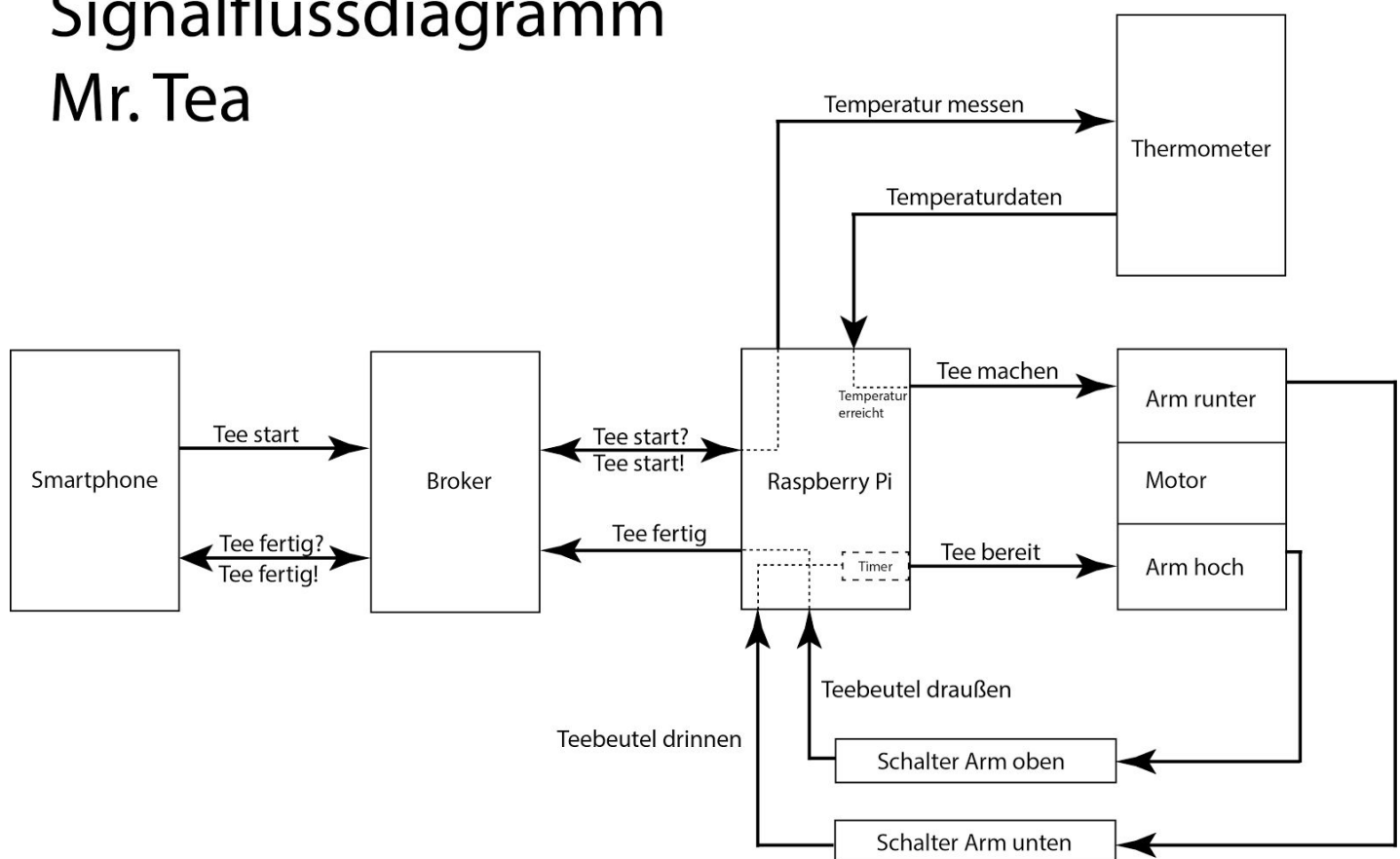
5) Technische Rahmenbedingungen (welche Hard- / Software)

- Javascript + HTML/CSS für Webapp und somit Android- und iOS- kompatibel
- Broker / Server zur Kommunikation zwischen Raspberry Pi und Smartphone
- Putty als Schnittstelle von Python und dem physikalischen Pins des Raspberry Pis
 - Skriptsprache über Jupyter Notebook für das Raspberry Pi ist Phyton
- Motor, der per H-Brücke in beide Richtungen bewegbar und an Batterie angeschlossen ist
- Zwei Schalter, die dem Raspberry Pi signalisieren in welcher Stellung der Arm ist
- Infrarot Thermometer sendet Wassertemperaturdaten an das Raspberry Pi

6) Technisches Konzept

Signalflussdiagramm

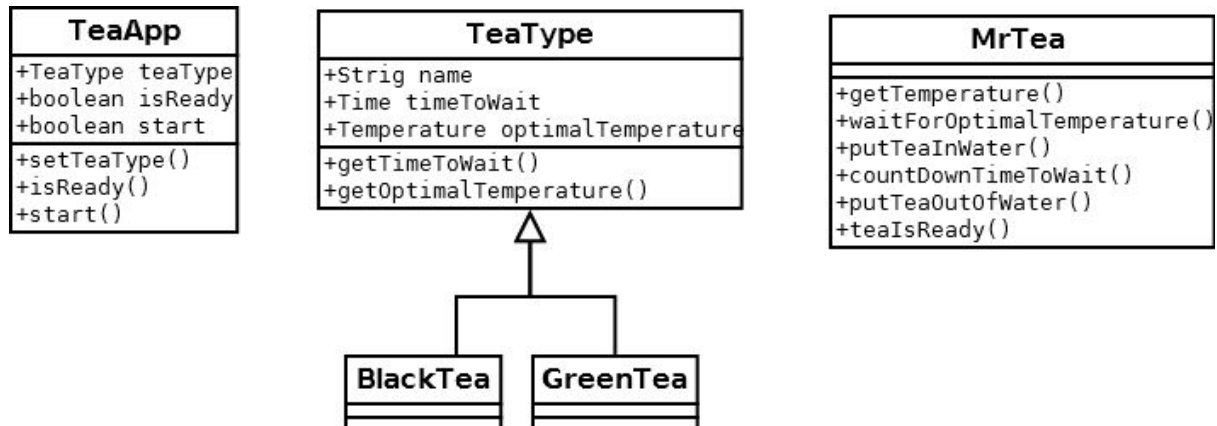
Mr. Tea



In dem dargestellten Signalflussdiagramm wird auf der rechten Seite des Brokers der Signalfluss von Mr. Tea skizziert. Dabei fragt der Raspberry Pi bei dem Broker an, ob er mit seinem Programm starten soll. Wird per Smartphone der Befehl gegeben, sendet der Raspberry Pi ein Signal an das Thermometer, das jetzt anfangen soll die Temperatur zu messen und zum Raspberry Pi zu übertragen. Liefert das Thermometer die geforderte Temperatur steuert der Raspberry Pi den Motor an, sodass dieser seinen Arm nach unten bewegt bis der Teebeutel im Wasser ist und der Arm einen Schalter betätigt. Dadurch wird dem Raspberry Pi ein Signal gesendet, dass der Tee nun im Wasser ist. Darauf wird ein Timer gestartet. Nachdem der Timer abgelaufen ist, wird wieder der Motor angesteuert, damit dieser nun den Arm in die andere Richtung fährt und damit den Tee aus dem Wasser

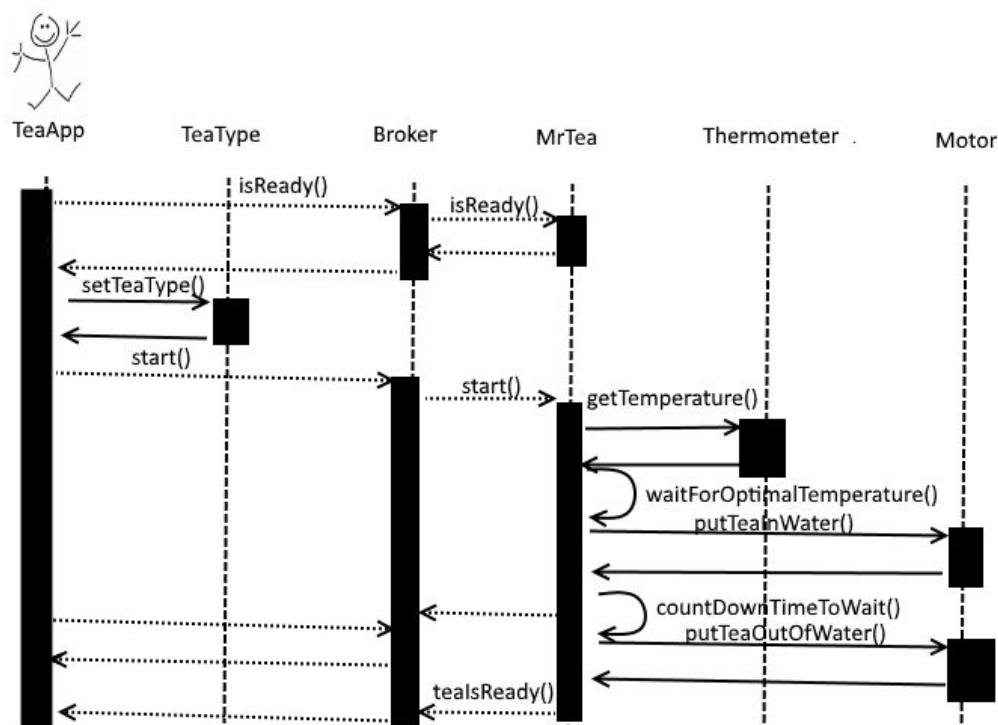
holt. Der Motor wird solange aktiviert bis er auf einen Schalter trifft, der dem Pi signalisiert, dass der Tee fertig ist.. Diese Information sendet der Raspberry Pi an den Broker, der dem Smartphone mitteilt, dass der Tee nun verzehrbereit ist.

UML Diagramme



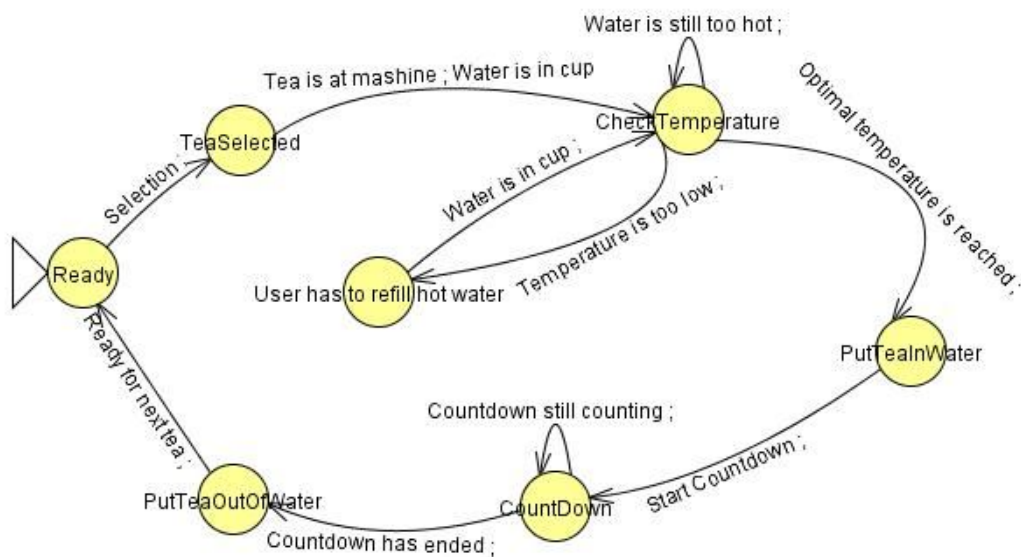
Die vorläufigen UML Diagramme stellen die Struktur der jeweiligen Klassen dar. Die Funktionen der Klasse *MrTea* werden auf dem Raspberry Pi implementiert und sind nicht an ein Objekt bzw. eine Klasse gebunden.

Sequenzdiagramm



Das Sequenzdiagramm soll einen Überblick für den zeitlichen Ablauf der Anwendung geben und die Zusammenhänge verdeutlichen. Der Broker wird sowohl mit der TeaApp, als auch mit dem Raspberry Pi (MrTea) über WLAN kommunizieren.

7) Bedienkonzept



Ein sehr abstraktes Modell der Zustände aus Sicht des Nutzers veranschaulicht die jeweiligen Schritte der Bedienung.

8) Team / Aufgabenverteilung

Artur Tahiraj	2009542	Hardware Engineer And Product Implementer
Stefanie Kohl	2011097	Design And Programming Executive Producer
Oliver Völling	2248958	Hardware And Software Gateway Engineer
Irena Becker	2238833	Lead Software Developer

9) Meilensteine und Aufwandschätzung

1. Git einrichten (1 Stunde) bis 09.05.
2. Materialliste (3 Stunden) bis 09.05.
3. Hardware ist funktionstüchtig (60 Stunden) bis 16.05.
4. Software ist geschrieben (30 Stunden) bis 23.05.
5. Funktionalität implementieren (Soft- & Hardware) (15 Stunden) bis 30.05.
6. Design der App / User Interface (10 Stunden) bis 20.06.
7. Feinschliff / eventuelle Zusatzfunktionen (30 Stunden) bis 20.06.
8. Generalprobe am 20.06.

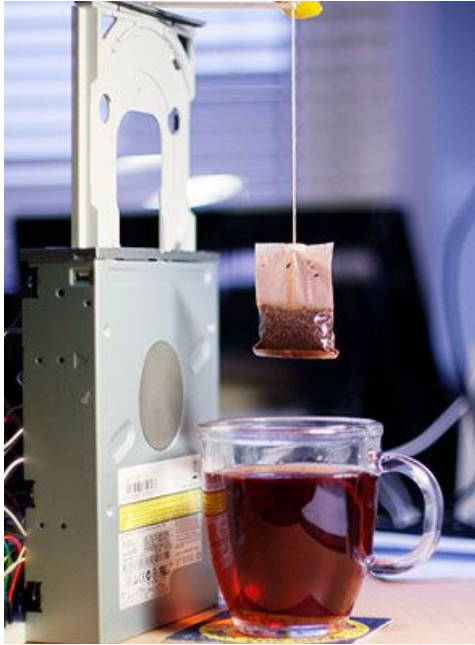
10) Inspiration

Abbildung 8: cdtea, <https://github.com/achilikin/cdtea>, aufgerufen am 29.04.17



Abbildung 9: Penelope, <http://www.modulo.co/projects/tea-brewing-robot>, aufgerufen am 29.04.17