

Projektplanung Smart-Bohner

Inhalt

| | |
|-----------------------------------|---|
| 1. Projektbeschreibung | 2 |
| 2. Rollen..... | 2 |
| 3. Funktionen..... | 2 |
| 3.1. Grundlegende Funktionen..... | 2 |
| 3.1.1. Kaffee machen..... | 2 |
| 3.1.2. Web UI..... | 2 |
| 3.2. Warnungen..... | 3 |
| 4. Technischer Rahmen | 3 |
| 4.1. Versionsverwaltung..... | 3 |
| 4.2. Projekt-Management | 3 |

1. Projektbeschreibung

Bei diesem Projekt handelt es sich um eine smarte Kaffee-Maschine, welche mit einem Raspberry-Pi angesteuert wird. Der Raspberry-Pi wird direkt an die Kaffee-Maschine angeschlossen und mit einer Web-API angesteuert.

2. Rollen

Die Rollen im Team sind folgendermaßen verteilt: Leon Burtscher übernimmt den Scrum-Master, allerdings ist er auch in der Entwicklung eingebunden. Felix Eckl und Michael Widerin übernehmen die Rolle als Entwickler.

3. Funktionen

In diesem Teil wird auf alle Funktionen eingegangen, die in diesem Projekt realisiert werden sollen, spätere Änderungen können nicht ausgeschlossen werden, wenn im Verlauf des Projekts Erkenntnisse gesammelt werden, aus welchen Funktionen ausgeschlossen werden müssen.

3.1. Grundlegende Funktionen

Dieser Teil fokussiert sich auf die Elementaren Funktionen, die in diesem Projekt umgesetzt werden müssen. Ohne diese kann das Projekt nicht fortgeführt werden, da dieser Teil die Elementaren Funktionen dieses Projekts darstellt

3.1.1. Kaffee machen

Als Benutzer möchte ich per API einen Kaffee machen können, ohne dabei eine Aktion an der Kaffee-Maschine ausführen zu müssen. Dieser wird über die Web-API gemacht und kann auch ohne UI möglich sein. Beim Kaffee machen spielt es anfangs keine Rolle, wie der Befehl an die API gesendet wird, da es in diesem Schritt um die Integration des Raspberry-PIs in die Kaffee-Maschine geht. Ein anfänglicher Schritt wäre es beispielsweise das Absenden eines POST-Befehls über Postman an die API: Bei Erhalt des Befehls sendet der Webserver den Befehl an die Pins sendet und den Knopfdruck an der Kaffee-Maschine „simuliert“.

3.1.2. Web UI

Als Benutzer möchte ich eine UI zur Verfügung haben, um die jeweiligen Befehle an die Web-API senden zu können. Die UI wird über Blazor realisiert, um eine einfache Anbindung an die API zu gewährleisten.

3.2. Warnungen

Bei der Kaffee-Maschine können diverse Warnungen und Fehler auftreten, welche am besten von der WEB-API ausgegeben werden sollten.

Als Benutzer möchte ich über bestimmte Events in der Web-Page informiert werden, um immer auf dem neuesten Stand sein. Wenn diese Meldung auftritt, kann der Benutzer keinen „Kaffee kochen“-Befehl mehr an die API senden. Wenn die Meldung wieder erfüllt wurde, dann verschwindet die Meldung und der Benutzer kann wieder Kaffee kochen. Folgende Meldungen sind dabei relevant:

- Wassertank füllen
- Kaffeesatzbehälter leeren

Neben diesen Meldungen, ohne welche der Benutzer keinen Kaffee kochen können, gibt es noch Meldungen, bei welchen es sich eher um „Warnungen“ handelt, da der Benutzer weiter Kaffee kochen kann, allerdings evtl. eingeschränkt.

- Geräte entkalken
- Gerät reinigen

3.3. Statistiken

Als Benutzer möchte ich über meine Statistiken Bescheid wissen. Ich möchte Statistiken über meinen täglichen Kaffee-Konsum erhalten. Diese Statistiken sind in der WEB-UI einsehbar

4. Meilensteine

Alle Meilensteine des Projekts sind auf Github dokumentiert

5. Technischer Rahmen

5.1. Versionsverwaltung

Bei dem Projekt wird Git als Versionsverwaltung gewählt und auf der Plattform Github verwaltet. Dabei wird jeder Commit/Pull-Request auf den „develop“ Branch mit einem Code-Review überprüft, um die Code-Sicherheit sicherstellt.

5.2. Projekt-Management

Das Projekt wird größtenteils über Github verwaltet. Auf Github gibt es die Möglichkeit, Tickets (Issues) anzulegen, darüber wird in diesem Projekt alle Fehler und Aufgaben verwaltet.