

Evaluation Sprint



Urban Green

Matthias Schwebler
Ramin Bahadoorifar
Samuel Schober
Konrad Kelc

	Name	Datum	Unterschrift
Erstellt:	M. Schwebler, R. Bahadoorifar	2.11.2016	
Geprüft:	S. Schober, K. Kelc	2.11.2016	

Contents

1	Einführung	4
2	Projektdaten	4
2.1	Projektteam	4
2.2	Projektbeschreibung	4
3	Voruntersuchung des Projekts	5
3.1	Ist-Erhebung	5
4	Aquaponic System	6
4.1	Design und Aufbau eines eigenen Systems	6
4.2	Verwendung des bereitgestellten Systems	7
4.3	Fazit	7
4.4	Abnahme	8
5	Single Board Computer	9
5.1	Arduino	9
5.2	Raspberry Pi B	9
5.3	Raspberry Pi Zero	10
5.4	Lösung	10
5.5	Abnahme	10
6	Sensoren	11
6.1	Temperatur	11
6.2	EC Wert	11
6.3	PH Wert	12
6.4	Abnahme	14
7	Aktoren	15
7.1	Pumpe	15
7.2	Futterautomat	17
7.3	Pflanzenbeleuchtung	17
7.4	Abnahme	18
8	Datenbankmanagementsystem	19
8.1	Evaluierung RDBMS vs. NoSQL	19
8.1.1	Einleitung	19
8.1.2	Kriterien	19

8.1.3	Optionen	19
8.1.4	Fazit	20
8.2	Evaluierung eines NoSQL Datenbanksystems	20
8.2.1	Einleitung	20
8.2.2	Kriterien	20
8.2.3	Optionen	21
8.2.4	Fazit	22
8.3	Gesamtfazit	22
9	Web Framework	23
9.1	Django	24
9.2	Node.js	25
9.3	Laravel	26
9.4	Fazit	27
10	Art der Datenübermittlung	29
10.1	Arduino zu Raspberry Pi und vice versa	29
10.1.1	JSON	29
10.1.2	CSV	29
10.1.3	Eigenes Format	29
10.1.4	Fazit	29
10.2	Raspberry Pi zu Server und vice versa	30
10.3	Abnahme	31
11	Fazit des kompletten Systems	32

Datum	Version	Autor	Status
26.10.2016	0.1	Matthias Schwebler, Ramin Bahadoorifar, Konrad Kelc, Samuel Schober	Erstentwurf
2.11.2016	0.2	Matthias Schwebler	Überarbeitung
16.11.2016	1.0	Matthias Schwebler, Ramin Bahadoorifar	Fertigstellung

1 Einführung

Aquaponik ist ein Verfahren, welches die Aufzucht von Fischen mit der Aufzucht von Nutzpflanzen verbindet. Im Wesentlichen handelt es sich dabei um einen geschlossenen Wasserkreislauf, in dem die entsprechenden Nährstoffe automatisch erzeugt werden; d.h. die Ausscheidungen der Fische werden durch Bakterien in Nährstoffe für die Pflanzen umgewandelt. Es soll eine vollautomatisierte Lösung für ein Aquaponik-System für kleine Haushalte als Prototyp geschaffen werden. Diese beinhaltet einerseits eine geeignete Konstruktion die sowohl das Aquarium enthält als auch die entsprechende Überwachung und Regelung des Systems.

2 Projektdaten

2.1 Projektteam

Name: Samuel Schober

E-Mail: sschober@student.tgm.ac.at

Name: Matthias Schwebler

E-Mail: mschwebler@student.tgm.ac.at

Name: Ramin Bahadoorifar

E-Mail: rbahadoorifar@student.tgm.ac.at

Name: Konrad Kelc

E-Mail: kkelc@student.tgm.ac.at

2.2 Projektbeschreibung

Im Rahmen des Projekts wird ein Tool zur Überwachung und Steuerung von Metadaten eines Aquaponic Systems, entwickelt und mit entsprechender Hardware realisiert. Dabei werden Sensoren von einem Raspberry Pi angesprochen und die Ergebnisse lokal gespeichert, welche später über eine Webseite bzw. einen Touchscreen auf dem Raspberry Pi eingesehen werden können. Des Weiteren kann der Benutzer über die Weboberfläche die Wassertemperatur sowie die Belichtungsdauer der Pflanzen regeln. Zusätzlich erhält der Benutzer Vorschläge für einen sinnvollen Besatz des Aquariums sowie für die Pflanzenwahl und deren Bedürfnisse.

3 Voruntersuchung des Projekts

3.1 Ist-Erhebung

Aquaponic Systeme sind hauptsächlich im größeren Maßstab bei der Aufzucht von Nutzpflanzen im Einsatz. Für den normalen Haushalt gibt es allerdings bis jetzt keine marktfähige Lösung. Die größten zwei Crowdfunding Projekte sind:

- **EcoQube C**

Der EcoQube C vereint ein handliches Aquaponics System mit elegantem Design, jedoch mangelt es an Konfigurierbarkeit. Es steht lediglich eine Fernbedienung zur Verfügung, mit der die Farbe der LEDs gesteuert werden kann. Des Weiteren liegt das Fassungsvermögen des Aquariums weit unter 50 Liter, was zur Folge hat, dass in Österreich maximal ein einziger Fisch darin gehalten werden kann. Daraus resultiert ein sehr kleines, ineffizientes Aquaponics System mit Mangel an Konfigurationsmöglichkeiten.

- **Grove Ecosystem**

Das Grove Ecosystem ist das "Non Plus Ultra", wenn es um Aquaponic Systeme im Haushalt geht. Es bietet alle möglichen Sensoren (Luftfeuchtigkeit und -temperatur sowie Wasserstand und -temperatur), welche über eine App abgefragt werden können. Diese bietet zusätzlich eine große Ansammlung an Daten und daher Empfehlungen für mögliche Fische und die dazu passenden Pflanzen.

Dieses Paket ist allerdings nur in den USA und Kanada, mit einem Einstiegspreis von > 4000€ erhältlich.

Beide dieser Systeme sind in Österreich kaum brauchbar bzw. nicht erhältlich. Bei Home Aquaponics wird Wert darauf gelegt, dass das fertige Produkt für jeden leistbar ist, indem Features weggelassen werden, welche nicht unbedingt benötigt werden. Außerdem wird äußerst stromsparende Hardware verwendet, um so wenig monatliche Kosten wie möglich zu verursachen.

4 Aquaponic System

Für die Verwendung unseres Systems, wird auch eine geeignete Konstruktion benötigt, in welcher unsere Hardware genügend Platz hat und wir die Testung aller Funktionalitäten durchführen können. Weiters soll die Aufbau-Art so kosteneffizient wie nur möglich gestaltet sein und nicht lange brauchen damit das vollständige System aufgebaut ist. Somit kommen wir auf die folgenden Kriterien:

- Design:
Gibt an, wie dynamisch das System aufgebaut ist .
- Kosten:
Gibt an, wie viel das System kostet
- Konstruktionsdauer:
Gibt an, wie lange es dauern würde bis das System aufgebaut ist

4.1 Design und Aufbau eines eigenen Systems

Design

Das Design eines, komplett auf uns angepassten, Systems ermöglicht es uns, die Hardware nach unseren Wünschen zu integrieren, sodass keinerlei fatale Probleme bei dem Aufbau unseres Systems auftauchen. Leider dauert das Design eines gut funktionierenden Systems viel Zeit und erfordert ein gewisses Maß an Erfahrung, in Sachen der Produktgestaltung.

Kosten

Die Kosten des Systems würden aufgrund der auf uns angepassten Bauteile und den speziellen Hardware-Teilen, einige hunderte Euro betragen dadurch wäre die Suche von Sponsoren nötig, was wiederum Zeit kostet, welche wir für die Fertigstellung des Projekts benötigen.

Konstruktionsdauer

Bei der Konstruktion eines eigenem Systems müssten wir, wegen mangelnder Erfahrung und Equipment, unsere Bauteile größtenteils outsourcen, somit würden wir sichergehen dass die Bauteile auch korrekt konstruiert werden, gehen aber das Risiko ein, dass von seitens des Herstellers, der Bauteile, ein Problem entsteht und wir die Testung unseres Systems verschieben müssen und dadurch eine unerwünschte Abhängigkeit entsteht.

4.2 Verwendung des bereitgestellten Systems

Design

Das fertige System, unseres Partners Ponix Systems, würde, aus designtechnischer Sicht, vollkommen für unseren Prototypen reichen. Es besitzt genug Platz, sodass unsere Hardware einen geeigneten Platz findet. Weiters bewirkt die Verwendung dieses Systems eine große Zeitersparnis unsererseits.

Kosten

Dadurch dass das System von unserem Partner Ponix Systems nicht mehr benötigt wird, hat Ponix Systems uns diese übergeben sodass wir damit unser Projekt durchführen können. Somit fallen vorerst keine Kosten an.

Konstruktionsdauer

Durch die fertige Konstruktion des Systems, müssen wir nur noch die Hardware in das System integrieren, dies bewirkt eine enorme Zeitersparnis bei der Konstruktionsdauer des Systems.

4.3 Fazit

Design und Aufbau eines eigenem Systems

Das Design und der Aufbau eines eigenem Systems ist nur wegen dem perfekt, auf uns, angepassten Design nicht die empfehlenswertere Wahl, weil ein schönes Design nur für die weitere Vermarktung wichtig ist und dies bei unserem Prototypen nicht vonnöten ist.

Verwendung des bereitgestellten Systems

Die Verwendung des bereitgestellten Systems ist die empfehlenswertere Wahl der beiden Möglichkeiten, weil das System so nicht nur gut genug für unseren Prototypen ist sondern wir uns auch noch vorerst alle Kosten sparen.

	Design	Kosten	Konstruktionsdauer
Design und Aufbau eines eigenen Systems	5/5	1/5	1/5
Verwendung des bereitgestellten Systems	4/5	5/5	4/5

4.4 Abnahme

Abnahme User Story 1826 - Evaluierung Aquaponik System		
Rolle	Name	Unterschrift
Autor	S. Schober	
Qualitätssicherung	K. Kelc	
Product Owner	S. Schober	

5 Single Board Computer

Um die Sensoren und Aktoren anzusprechen wird ein SBC (Single Board Computer) verwendet. Die Anzahl an erhältlichen SBCs, wird der Fokus auf populäre Produkte mit einer großen Community gelegt. Ideal wäre ein SBC, der auf Messgeräte, wie sie im Aquarium verwendet werden, spezialisiert ist und ein WLAN-Interface, für die Datenübermittlung hat.

5.1 Arduino

Der Arduino Uno ist ein SBC, welcher im Bereich von 20 - 25 € zu erhalten ist. Er hat einen internen Speicher der groß genug ist, um einen Netzausfall überbrücken zu können. Da er aber selbst keine Netzwerkschnittstelle hat, muss er mit einem WiFi-Shield (€88) ausgestattet werden. Ein Shield ist eine Hardware-erweiterung, die dem Arduino situationsbezogene Funktionen hinzufügen kann. Für dieses Projekt ist das "Open Aquarium Board" (€50) optimal, da es speziell entwickelt wurde um ein Aquarium zu überwachen bzw. zu steuern und eine Open Source API anbietet.

Das Problem dabei ist, dass nur ein Shield auf dem Arduino angebracht werden kann. Das heißt, entweder das Open Aquarium Board wird verwendet, oder ein WLAN Shield zur Datenübermittlung. Er müsste mit einem anderen SBC, welcher Netzwerktauglich ist, verbunden werden um die Daten auf einem zentralen Server zu persistieren.

5.2 Raspberry Pi B

Der Raspberry Pi B, ist einer der energieeffizienteren Computer seiner Art (1.8V im Leerlauf). Er bietet ebenso wie der Arduino einen adäquaten Speicher, kann aber einfach mit der eingebauten RJ45 Buchse bzw. einem WLAN-Stick mit dem Internet verbunden werden. Ein solcher WLAN-Stick ist mit €10 im Vergleich zu dem WiFi-Shield billiger. Da aber der Raspberry Pi B keine API für jegliche Sensoren und Aktoren bietet, erfordert die Implementierung der Überwachung bzw. Steuerung des Aquaponic Systems einen extremen Arbeitsaufwand.

5.3 Raspberry Pi Zero

Ebenso wie der Raspberry Pi B, hat der Pi Zero einen internen Speicher und kann mit einem WLAN-Stick mit einem Lokalen Netzwerk verbunden werden. Der Unterschied ist, dass er keine RJ45 Buchse besitzt und um einiges weniger Rechnerleistung. Der Vorteil dabei ist, dass er extrem wenig Strom verbraucht (0.45V im Leerlauf).

5.4 Lösung

Aus Erfahrungswerten der Firma "Ponix Systems", ist das Open Aquarium Board für den Arduino unabdinglich. Der optimale Weg ist also, ein Arduino mit entsprechendem Shield, der die Daten an einen Raspberry Pi übermittelt, welcher diese dann an einen Server schickt.

5.5 Abnahme

Abnahme User Story 1817 - Single Board Computer		
Rolle	Name	Unterschrift
Autor	M. Schwebler	
Qualitätssicherung	R. Bahadoorifar	
Product Owner	S. Schober	

6 Sensoren

Die Sensoren für den Prototypen, werden von der Firma Ponix Systems zur Verfügung gestellt. Diese müssen allerdings erst auf Funktionalität untersucht werden.

Für die Entwicklung, des Codes, zum Ansprechen der Sensoren, wird "Arduino IDE" verwendet. Diese IDE bietet die Möglichkeit, den Code auf einem rechenstarken PC zu kompilieren und ihn dann auf dem Arduino auszuführen. Zusätzlich kann sie verwendet werden, um alle eingehenden Daten des Arduinos zu überwachen. Der kompilierte Code wird beim Endprodukt auf dem Arduino gespeichert und sendet kontinuierlich Daten, über die USB Schnittstelle, an den Raspberry Pi.

Es werden Libraries von Arduino indirekt über die API von OpenAquarium verwendet.

6.1 Temperatur

Die Temperatur kann mit nur einem Befehl abgelesen werden. Dieser liefert einen Wert in Grad Celsius zurück.

```
OpenAquarium.init();  
float temperature = OpenAquarium.readtemperature();
```

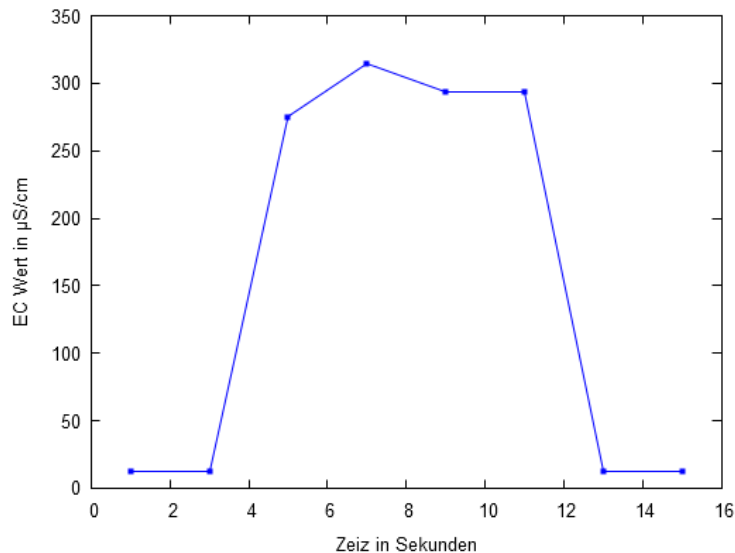
Die Testergebnisse des Temperatur Sensors stimmten mit denen eines handelsüblichen Thermometers überein.

6.2 EC Wert

Der EC Wert gibt Aufschluss darüber, wie viele Salze jeglicher Art im Wasser gelöst sind. Desto mehr Nährstoffe also im Wasser vorhanden sind, umso höher ist auch der EC Wert.

```
OpenAquarium.init();  
OpenAquarium.calibrateEC(point_1_cond, point_1_cal,  
                          point_2_cond, point_2_cal);  
  
float resistanceEC = OpenAquarium.readResistanceEC();  
float EC = OpenAquarium.ECConversion(resistanceEC);  
      //EC Value in uS/cm
```

Die Tests beim Eintauchen des Sensors in Leitungswasser ergaben folgende Ergebnisse:



Wie erwartet, steigt der EC Wert beim Eintauchen ins Wasser, da es natürlich um einiges leitfähiger ist, als Luft.

6.3 PH Wert

Allgemein

Der PH Wert ist einer der wichtigsten Werte, wenn es um das Überwachen eines Aquaponik Systems geht. Dafür gibt es ein großes Spektrum an erhältlichen Sensoren. Die der unteren Preisklasse, benötigen eine regelmäßige Wartung. Die der oberen Preisklasse heben den Preis des Produktes, über die Zielgruppe hinaus, und sind deshalb unbrauchbar.

Da der Sensor ein analoges Signal liefert, muss die gemessene Spannung erst umgerechnet werden. Dies übernimmt die Klasse OpenAquarium.

```
OpenAquarium.calibratepH(cp4, cp7, cp10);  
int mvpH = OpenAquarium.readpH(); //Value in mV of pH  
float pH = OpenAquarium.pHConversion(mvpH);
```

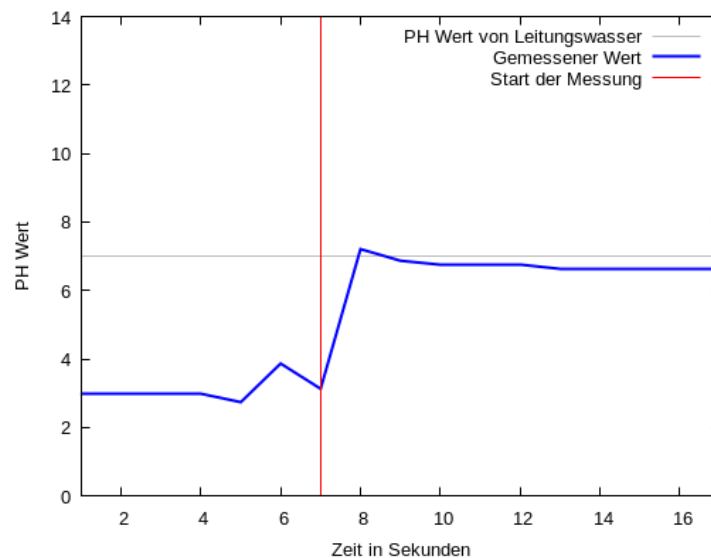
cp4, cp7, cp7 ... Calibrationpoints (Konstanten)

Calibrationpoints

Calibrationpoints (Kalibrierungswerte), werden angepasst, wenn der gemessene PH Wert zu stark vom eigentlichen Wert abweicht. Um die Abweichung des Sensors zu beheben, sind Kalibrierungskits vorgesehen, die allerdings bis zu ein mal im Monat verwendet werden müssen. Dies belastet den Kunden mit zusätzlichen Kosten und Aufwand und soll daher umgangen werden.

Tests

Beim testen eines bereits länger in Betrieb gewesenem PH Sensor brachte dieser folgende Ergebnisse:



Wie zusehen ist, weicht der gemessene Wert bereits stark von dem eigentlichen Wert ab. In einem Aquaponik System hat eine Abweichung dieser gröÙe, starke Auswirkungen auf das Gleichgewicht des Ecosystems.

Fazit

Da es keine PH Sensoren gibt, die billig sind und zugleich einen niedrigen Wartungsaufwand aufweisen, wird je nach Möglichkeiten versucht eine eigene Lösung zu etablieren. Dies soll mithilfe einiger Teststreifen und einer Kamera passieren, bei der, der Benutzer per Knopfdruck über die Website den PH Wert messen kann. Die Vorteile bei dieser Lösung sind, dass solch ein System einzigartig auf dem Markt wäre, und dass der Benutzer selbst bestimmen kann, ob er den PH Wert messen will und wie oft. Wie genau dieses System realisiert wird, wird im nächsten Sprint (Sprint 1) geplant.

6.4 Abnahme

Abnahme User Story 1975 - Sensoren		
Rolle	Name	Unterschrift
Autor	M. Schwebler	
Qualitätssicherung	R. Bahadoorifar	
Product Owner	S. Schober	

7 Aktoren

Es werden insgesamt vier Aktoren von Ponix Systems zur Verfügung gestellt. Darunter befinden sich: 2 Pumpen (Wasser/Mineralien), 1 Futterautomat und eine Pflanzenbeleuchtung.

Um dem Kunden die Arbeit abzunehmen die Zimmertemperatur so zu regulieren um ein Klima für die Fische zu schaffen, wird zusätzlich ein Heizstrahler verbaut. Dieser wird im nächsten Sprint (Sprint 1) geplant und eingesetzt.

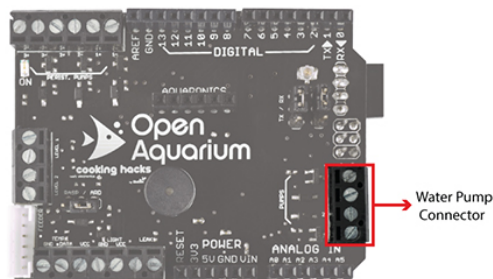
7.1 Pumpe

Wasser

Die Wasserpumpe kann mit einer Zeile Code ein bzw. ausgeschaltet werden. Die Steuerung, die kontrolliert wann genau die Pumpe eingeschalten wird, muss erst implementiert werden (Sprint 1).

```
OpenAquarium.pumpON(1);  
OpenAquarium.pumpOFF(1);
```

Zum Anschließen der Pumpe sind auf dem AquariumBoard Steckplätze vorgesehen.



Pumpleistung: 100-350 L/H

Spannung: 3.5-12 DC

Leistungsbereich: 0.5W-5W

Mineralien

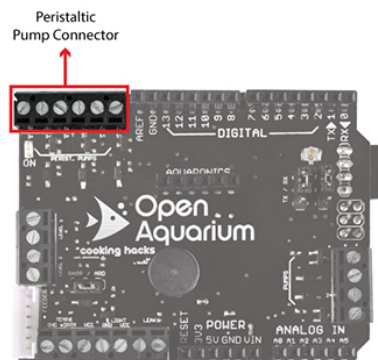
Wie bei der Wasserpumpe ist der Code zum Aktivieren und Deaktivieren der Pumpe kurz.

```
OpenAquarium.perpumpON(1);  
OpenAquarium.perpumpOFF(1);
```

Die Dosierung der Mineralien muss nicht manuell vorgenommen werden sondern kann über das OpenAquarium.h File angepasst werden.

```
#define dosingdrop1 5  
#define dosingdrop1 10  
#define dosingdrop1 15
```

Zum Anschließen der Pumpe sind wieder Steckplätze vorhanden.



12V DC input voltage
Flow: 20-60 ml/min

7.2 Futterautomat

Um den Wartungsaufwand zu minimieren, ist ein Futterautomat notwendig, der über eine Weboberfläche konfiguriert werden kann. Es kann entweder ein teurer Futterautomat gekauft werden, der diese Option anbietet, oder ein durchschnittlicher Futterautomat wird so bearbeitet, dass er auf ein Stromsignal reagiert indem er Futter ausgibt.

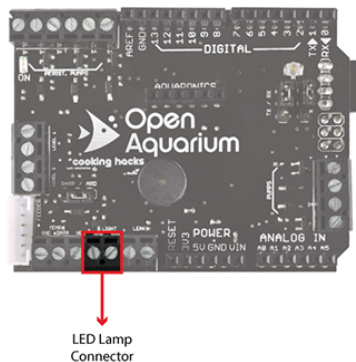
Der Futterautomat von Ponix Systems gehört zur letzteren Art. Sobald 5 Volt angelegt wird, gibt er Futter aus. So kann später über die WebApp das Intervall und die Menge gesteuert werden.

7.3 Pflanzenbeleuchtung

Die Beleuchtung wird mit einem LED Streifen realisiert von www.cooking-hacks.com (bzw. von Ponix Systems gesponsort) realisiert. Er besteht aus 72 LEDs (60 Weiße und 12 Blaue) mit einer Leuchtkraft von 500 Lumen. Der Streifen kann mit Hilfe der OpenAquarium Klasse der aktuellen Zeit automatisch angepasst werden.

```
now = OpenAquarium.getTime();  
OpenAquarium.printTime(now);  
OpenAquarium.lighting(now);
```

Wie bei den beiden Pumpen kann der LED Streifen am OpenAquarium Board angeschlossen werden.



Leistung: 5.5 watts

Beleuchtung: 72 LEDs

Farbe: 60 Weiße + 12 Blaue)

7.4 Abnahme

Abnahme User Story 1979 - Aktoren		
Rolle	Name	Unterschrift
Autor	Matthias Schwebler	
Qualitätssicherung	Konrad Kelc	
Product Owner	Samuel Schober	

8 Datenbankmanagementsystem

8.1 Evaluierung RDBMS vs. NoSQL

8.1.1 Einleitung

Zur Speicherung der Sensordaten soll ein geeignetes Datenbankmanagementsystem verwendet werden. Zu Beginn wird evaluiert, ob ein relationales oder ein NoSQL-Datenbanksystem verwendet wird. Dabei werden die jeweiligen Vor- und Nachteile von SQL und NoSQL Datenbanken aufgelistet. Besonders wert gelegt wird auf die Performance, die Konsistenz sowie dem einfachen Zugriff auf die Daten.

8.1.2 Kriterien

- Performance:
Gibt an, wie schnell die Lese- und Schreibleistung ist.
- Konsistenz:
Gibt an, wie gut das DBMS Maßnahmen zur Konsistenzerhaltung umsetzt.
- einfacher Zugriff auf Daten:
Gibt an, wie umfangreich die API des DBMS ist.

8.1.3 Optionen

SQL-Datenbanken Abfragesprache

Die Abfragesprache SQL wird auf die drei Teile DDL(Data Definition Language), DML(Data Manipulation Language) und DCL (Data Control Language) aufgeteilt.

- DDL:
ist die Sprache für das Anlegen, Ändern und Löschen von Datenstrukturen.
- DML:
ist die Sprache für das Abfragen, Einfügen, Ändern oder Löschen von Nutzdaten.
- DCL:
ist die Sprache für die Zugriffskontrolle.

Konsistenz

SQL-Datenbanken verwenden das ACID Prinzip, daher besitzen sie eine bessere Konsistenz als NoSQL-Datenbanken. NoSQL Datenbanken verwenden das BASE Prinzip, die Daten hier sind daher nur "schlussendlich" konsistent. Dadurch kann es passieren das nach einem Update nicht immer die selben Werte zurückgeliefert werden.

NoSQL-Datenbanken NoSQL-Datenbanken besitzen wegen ihres einfachen Schemas eine hohe Skalierbarkeit, welche trotz eines hohen Datenvolumens gewährt wird.

8.1.4 Fazit

SQL-Datenbanken weisen eine bessere Konsistenz auf, jedoch besitzen NoSQL-Datenbanken eine bessere Performance und Skalierbarkeit. Da wir besonderen Wert auf die Performance legen, haben wir uns entschieden eine NoSQL-Datenbank zu verwenden.

8.2 Evaluierung eines NoSQL Datenbanksystems

8.2.1 Einleitung

Die Folgenden Datenbankmanagementsysteme wurden evaluiert:

- MongoDB
- Redis

8.2.2 Kriterien

- Performance:
Gibt an, wie schnell die Lese- und Schreibleistung ist.
- einfacher Zugriff auf Daten:
Gibt an, wie umfangreich die API des DBMS ist.
- Dokumentation:
Gibt an, wie gut das DBMS dokumentiert ist.
- Verbreitung:
Gibt an, wie weit das DBMS verbreitet ist.

8.2.3 Optionen

MongoDB MongoDB ist eine Schema-freie, dokumentenorientierte NoSQL-Datenbank, die in der Programmiersprache C++ geschrieben ist. Da die Datenbank dokumentenorientiert ist, kann sie Sammlungen von JSON-ähnlichen Dokumenten verwalten. So können viele Anwendungen Daten auf natürlichere Weise modellieren, da die Daten zwar in komplexen Hierarchien verschachtelt werden können, dabei aber immer abfragbar und indizierbar bleiben.

Außerdem ist MongoDB eine Allzweckdatenbank mit offenem Quellcode (OpenSource). Sie hat u.a. folgende Merkmale:

- Dokumentdatenmodell mit dynamischen Schemata
- Umfassende, flexible Indexunterstützung
- Aggregation-Framework und MapReduce
- Auto-Sharding für horizontale Skalierbarkeit
- Eingebaute Replikation für Hochverfügbarkeit
- Textsuche
- Erweiterte Sicherheit (z.B. Kerberos Unterstützung)
- Speicherung großer Dateien mit GridFS

Redis Redis ist eine In-Memory-Datenbank mit einer einfachen Schlüssel-Werte-Datenstruktur. Nach einer Erhebung von DB-Engines.com ist Redis der verbreitetste Schlüssel-Werte-Speicher.

Die einfache Struktur der Datenbank eignet sich weniger für komplexe Datenstrukturen, die überwiegend in der Datenbank selbst abgebildet werden soll, dafür ist der große Vorteil von Redis, dass es schneller ist als relationale Datenbanken wie z.B. MySQL. Bis zu ca. 100.000 Schreibvorgänge und ca. 80.000 Lesevorgänge pro Sekunde sind auf herkömmlicher Hardware möglich.

Redis bietet Persistenz durch automatisiertes regelmäßiges Abspeichern oder per Protokolldatei, wodurch bei entsprechender Konfiguration auch eine ACID-konforme Dauerhaftigkeit erreichbar ist.

8.2.4 Fazit

Redis schneidet bei den Kriterien, wie z.B. der Performance und den einfachen Zugriff auf Daten durch seine Kompaktheit, besser als MongoDB ab, hat jedoch den großen Nachteil, dass es bei serverseitiger Verwendung, extrem viel Arbeitsspeicher verbraucht, wenn einige Aquaponik Systeme ihre Daten über den Server persistieren.

8.3 Gesamtfazit

Es müssen sowohl auf dem lokalen Raspberry Pi, als auch auf dem Server Echtzeitdaten vorhanden sein, um einerseits auf dem Touchdisplay eine Übersicht anzuzeigen und andererseits über den Browser alle Daten zu betrachten. Da auf dem Raspberry Pi allerdings nicht viel Speicherplatz aber genügend Arbeitsspeicher vorhanden ist, wird dafür Redis verwendet und für die serverseitige Persistierung MongoDB.

Abnahme User Story 1819 - Datenbankmanagementsystem		
Rolle	Name	Unterschrift
Autor	Konrad Kelc	
Qualitätssicherung	Matthias Schwebler	
Product Owner	Samuel Schober	

9 Web Framework

Bei Web Frameworks, werden durch vordefinierte und vorgefertigte Klassen, häufig gebrauchte Funktionen wie Mailversand, sichere Authentifizierung, Sicherheitsfunktionen, Lokalisierung, Performance (z.B. HTTP Caching) oder grundlegende Funktionen für Webformulare vereinfacht.

Web Frameworks sind darauf ausgelegt, sehr schnell Ergebnisse zu erzielen und lauffähige Webanwendungen zu erstellen. Dazu bieten sie einen Datenbankzugriff, Templating-Mechanismen, eine saubere Trennung von Präsentation und Code durch Verwendung des Model-View-Controllers.

Da eine eigenständige Implementierung all dieser Mechanismen den Arbeitsaufwand übersteigen würde, ist eine Verwendung eines solchen Web Frameworks unabdinglich. Welches Framework dabei gewählt wird hängt von folgenden Komponenten ab:

- Schwierigkeitsgrad
- Lernkurve
- Sicherheit
- Dokumentation

9.1 Django

Django ist ein in Python geschriebenes opensource Web Application Framework, das einem Model-View-Presenter-Schema folgt.

Schwierigkeitsgrad/Lernkurve

Da Python selbst leicht zu erlernen ist, ist es keine Herausforderung sich in die Sprache einzuarbeiten. Das Framework ansich, gilt als eines der einfacheren (aber auch funktionsärmeren) seiner Art und bietet daher eine flache Lernkurve. Für jemanden der bereits mit Webframeworks gearbeitet hat, ist Django schnell zu erlernen.

Sicherheit

Django ist so konzipiert, dass die standardmäßig implementierten Sicherheitsmaßnahmen, für den größten Teil der Webentwicklung ausreicht. Da zwischen dem Raspberry Pi und dem Server wenig sicherheitsrelevante Daten ausgetauscht werden, genügt die Sicherheit von Django.

Dokumentation

Die Dokumentation von Django ist gerade für Einsteiger optimal, da sie einige nützliche Tutorials bietet. Aber auch für erfahrene Entwickler ist die Dokumentation großartig.

9.2 Node.js

Node.js ist eine Event-basierte Plattform, welche nicht blockierend ist. Das bedeutet, dass der Server nicht wartet bis ein Ereignis vollendet wurde, um mit den nächsten Aufgaben fortfahren zu können. Stattdessen werden Callbacks verwendet, welche dann erst aufgerufen werden, wenn ein bestimmtes Ereignis fertig ausgeführt wurde, bei dem von einer längeren Dauer ausgegangen wird. Währenddessen kann der Server andere Aufgaben bearbeiten.

An sich läuft Node.js nicht Multi-Threaded, wie man vielleicht erwartet, sondern bearbeitet jede eingehende Verbindung mit einem Thread. Jedoch werden bestimmte Funktionen, wie das Auslesen einer Datenbank oder das Rendern von Templates, intern durch mehrere Threads erledigt, welche dann bei Vollendung den zugewiesenen Callback aufrufen. Durch diese Architektur ist Node.js in vielen Bereichen viel schneller als andere Webserver, die für jede Verbindung einen Prozess erstellen würden.

Node.js ist sehr abhängig durch Third-Party Module, welche durch NPM (Node Package Manager) verwaltet und installiert werden können. Diese Module machen das Entwickeln von Applikationen viel einfacher oder geben auch mehr Funktionen. Für uns interessante Module wären zum Beispiel: Express.js oder Socket.io. Ersteres gibt uns die Möglichkeit die Applikation im MVC-Pattern zu entwickeln. Mit Socket.io könnte man Daten dem Client senden und gleichzeitig empfangen, ohne dabei davor eine Anfrage vom Client erwarten zu müssen. Es genügt damit nur noch eine Anfrage vom Client bei dem eine Verbindung (Socket) zum Server geöffnet wird, sodass man Daten bidirektional senden kann.

Schwierigkeitsgrad/Lernkurve

Wie der Name schon verrät verwendet Node.js JavaScript als Programmiersprache. Falls man die Sprache schon kann, wie in unserem Fall, ist eine Lernkurve jeweils in den einzeln verwendeten Modulen sichtbar, die sehr unterschiedlich sind. Somit ist der Schwierigkeitsgrad nicht wirklich einschätzbar. Der Schwierigkeitsgrad in den oben genannten Modulen liegt im mittleren Bereich.

Sicherheit

Node.js hat eine relativ große Community. Das ist insofern wichtig, da Sicherheitslücken dadurch schneller gefunden und behoben werden können. Jedoch unterscheidet sich die Sicherheit auch in diesem Punkt in den einzelnen Modulen. Somit muss beachtet werden, dass der zu verwendende Modul eine hohe Bekanntheit hat.

Dokumentation

Node.js ist an sich sehr gut dokumentiert und bietet durch eine Vielzahl von Tutorials einen einfachen Einstieg. Von den bekannten Modulen, wissen wir, dass die Dokumentation sehr gut ist, wie bei Socket.io oder Express.js. Außerdem wird der Einstieg in diesen Modulen mit verschiedenen Tutorials, die im Internet zu finden sind, sehr einfach gemacht.

9.3 Laravel

Laravel ist ein PHP-Framework und wird von vielen Entwickler derzeit als eines der besten Frameworks für PHP bezeichnet. Laravel bietet viele Funktionen an, wie Routing, Templating, Unit Testing, usw.

Die Architektur von PHP unterscheidet sich von Node.js sehr stark:

- Pro Verbindung wird ein neuer Prozess erstellt. Dies hat bei hoher Nutzzahl zur Folge, dass der Arbeitsspeicher des Servers sehr schnell voll werden kann. Node.js verwendet nur einen Thread und erstellt, falls erforderlich, mehrere Threads für eine jeweilige Aufgabe, welche dann bei erfolgreicher Ausführung (also Aufrufen des Callbacks) beendet werden. Wird aber ein Fehler hervorgerufen stürzt Node.js komplett ab, wogegen bei PHP nur der Prozess abstürzt.
- Jedesmal, wenn ein PHP-Skript ausgeführt wird, wird das Skript neu interpretiert. Dies hat den Vorteil, dass bei einer Codeänderung nicht der Server neugestartet werden muss. Der Nachteil liegt jedoch in der Performance. Bei Node.js wird der Code beim Starten kompiliert. Eine Codeänderung wirkt sich jedoch solange nicht am Server aus bis Node.js neugestartet wurde. Dies dauert deutlich länger als ein PHP-Server. Dagegen ist die Performance in der Laufzeit deutlich schneller als bei PHP.
- Der Code in PHP läuft, anders als in Node.js, synchron ab. So muss zum Beispiel für eine Datenbankverbindung solange gewartet werden, bis diese auch erfolgreich erstellt worden ist. Dies würde die Antwortzeit einer Anfrage sehr beeinträchtigen.

Schwierigkeitsgrad/Lernkurve

Die Skriptsprache PHP ist sehr einfach zu lernen, genauso wie das Framework Laravel. Jedoch haben die Teammitglieder nicht viel Erfahrung mit PHP, sodass etwas Zeit für das Lernen vergehen würde.

Sicherheit

Die Standard implementationen und Sicherheitsmaßnahmen sind für den größten Teil der Webentwicklung ausreichend und decken die Ansprüche des Projekts vollständig.

Dokumentation

PHP ansich bietet eine hervorragende Dokumentation mit einigen Beispielen, was den Einstieg in die Sprache erleichtert. In die Dokumentation von Laravel muss man allerdings viel Zeit investieren um sich einzulesen.

9.4 Fazit

Django

Für Einsteiger ist Django definitiv empfehlenswert. Die gute Dokumentation ermöglicht es jeden Entwickler mit - egal mit welchen Kenntnissen - sich einen guten Überblick, was für dieses Projekt bereits ausreichend ist.

Node.js

Node.js kann mit seinen Modulen innovative Lösungen bieten, wie das schon erwähnte Socket.io-Modul. Außerdem besitzt Node.js eine sehr große Community, die ständig wächst, wie auch der derzeitige Trend zeigt. Große Unternehmen, wie Netflix oder Paypal, basieren auf Node.js, aufgrund seiner Schnelligkeit und hohen Skalierbarkeit.

Laravel

PHP bzw. Laravel ist für unsere Anforderungen nicht besonders gut geeignet, da unser Produkt ein schnelles System erfordert, sowie eine hohe Zahl von Verbindungen. Dies würden wir mit PHP nicht erreichen können.

Bewertung der Frameworks nach Schulnotensystem (1 - Sehr gut bis Nicht genügend - 5)

Framework	Schwierigkeit	Sicherheit	Dokumentation	Erfahrung
Django	2	3	2	2
NodeJS	1	2	1	1
Laravel	3	2	2	3

Abnahme User Story 1818 - Webframework		
Rolle	Name	Unterschrift
Autor	Ramin Bahadoorifar, Matthias Schwebler	
Qualitätssicherung	Konrad Kelc	
Product Owner	Samuel Schober	

10 Art der Datenübermittlung

Bei der Datenübertragung sind mehrere Parameter zu beachten. Dazu gehören hauptsächlich das Format der Daten und das Intervall in denen diese übertragen werden.

10.1 Arduino zu Raspberry Pi und vice versa

Um die Sensordaten dem Raspberry Pi mitzuteilen, eignet sich am besten ein Key-Value Format. Dabei gibt es verschiedene Möglichkeiten, wie z.B. JSON und CSV.

Das Intervall der Datenübertragung kann frei, vom Benutzer gewählt werden.

10.1.1 JSON

JSON (JavaScript Object Notation) hat den großen Vorteil, der leichten Server-seitigen Auswertung und Konvertierung in ein Javascript Object. Da die Daten aber vorübergehend auf dem Raspberry Pi persistiert werden sollen, und Redis kein automatisches speichern von JSON Objekten unterstützt erzeugt diese Methode der Übertragung mehr Aufwand als Nutzen.

10.1.2 CSV

Ähnlich wie bei JSON, müssten die Daten erst bearbeitet und gefiltert werden, um in die Datenbank gespeichert werden zu können.

10.1.3 Eigenes Format

Um die Daten mit Redis zu persistieren wird folgende Syntax verwendet: "Key1 Value1 Key2 Value2 ...". Um Mehraufwand durch Verwendung eines standardisierten Datenformats zu vermeiden, können die Daten gleich so gesendet werden, dass sie ohne Bearbeitung in Redis gespeichert werden können.

10.1.4 Fazit

Der Datenaustausch wird in folgendem Format stattfinden:

<code>temperature 23.4 ec 285.7</code>
--

10.2 Raspberry Pi zu Server und vice versa

Wenn die Daten in Redis gespeichert worden sind, müssen diese dem Zentralserver gesendet werden. Der Sender auf dem Raspberry Pi ist Node.js, wie auf dem Empfänger bzw. Server.

Die Daten schauen bei Auslesen der Datenbank wie folgt aus:

- 1) temperature
- 2) 23.4
- 3) ec
- 4) 285.7

Wie wir in Kapitel 10.1 festgestellt haben, offenbart sich die Übertragung in JSON nicht nur wegen der Einfachheit der Konvertierung zu einem Javascript Objekt als Vorteil, sondern auch aus drei anderen Gründen:

- **Einfache Speicherung in MongoDB:** Da MongoDB JSON als Speicherformat verwendet, ist es deswegen für uns einfacher JSON-Objekte in die Datenbank zu speichern.
- **Übertragung der Daten durch Socket.io:** Daten werden standardmäßig in Socket.io als JSON-Objekt gesendet.
- **Oberfläche der Webapplikation durch Angular:** Angular bietet Funktionen an, JSON-Objekte in der Weboberfläche anzeigen zu lassen. Falls sich das JSON-Objekt verändert, wird automatisch auch die Oberfläche dementsprechend angepasst.

Deswegen werden die Daten wie folgt ausschauen:

```
{  
  temperature: "23.4",  
  ec: "285.7"  
}
```

10.3 Abnahme

Abnahme User Story 2049 - Art der Datenübermittlung		
Rolle	Name	Unterschrift
Autor	Matthias Schwebler, Ramin Bahadoorifar	
Qualitätssicherung	Konrad Kelc	
Product Owner	Samuel Schober	

11 Fazit des kompletten Systems

