

Pflichtenheft

Projektbezeichnung	Softwareprojekt OnTop
Projektleiter	Henrick Blunck, Stefan Jonker, Marc-Maurice Torchala
Erstellt am	06.05.2021
Letzte Änderung am	06.05.2021
Status	in Bearbeitung
Aktuelle Version	1.0

Änderungsverlauf

Nr.	Datum	Version	Geänderte Kapitel	Art der Änderung	Autor	Status
1	-	-	-	-	-	-

Inhalt

1. Einleitung	3
2. Allgemeines	3
2.1. Ziel und Zweck des Dokuments	3
2.2. Ausgangssituation	3
2.3. Projektbezug	3
2.4. Abkürzungen	3
2.5. Teams und Schnittstellen	3
3. Konzept	4
3.1. Ziel(e) des Anbieters	4
3.2. Ziel(e) und Nutzen des Anwenders	4
3.3. Zielgruppe(n)	5
4. Funktionale Anforderungen	5
5. Nichtfunktionale Anforderungen	5
5.1. Allgemeine Anforderungen	6
5.2. Gesetzliche Anforderungen	6
5.3. Technische Anforderungen	6
6. Rahmenbedingungen	6
6.1. Zeitplan	6
6.2. Technische Anforderungen	7
6.3. Problemanalyse	7
6.4. Qualität	8
7. Liefer- und Abnahmebedingungen	8
8. Anhang	8

1. Einleitung

Das vorliegende Pflichtenheft enthält die an das Softwareprojekt gestellten funktionalen sowie nicht-funktionalen Anforderungen. Es dient als Basis für die Anforderungen und Umsetzung. Aufgelistet werden alle, für die Umsetzung des Projektes, relevanten Schritte, um zu einem, für alle Beteiligten, zufriedenenden Ergebnis zu gelangen.

2. Allgemeines

2.1. Ziel und Zweck des Dokuments

Dieses Pflichtenheft beschreibt das Vorgehen unseres Teams zur Automatisierung und Auswertung durch Sensorik des Interdisziplinären Projekts der OnTop Rooftop Farm der Hochschule Bochum.

2.2. Ausgangssituation

Das neue Informatikteam des Projekts OnTop Bochum hat sich dem bereits bestehendem kompletten Projektteam angeschlossen und besteht aus Lucas Degenhardt-Seltmann, Florian Rüßler, Talha Taskaya und Tobias Maximilian Alexandre Basten.

2.3. Projektbezug

Das vorliegende Projekt ist ein Teilprojekt der OnTop Rooftop Farm welches von Studenten der Nachhaltigen Entwicklung 2019 gestartet wurde. Es hat zum Ziel eine Ökologische und Nachhaltige Nutzung von sonst ungenutzten Dachflächen zu erforschen. Unser Teil des Projektes befasst sich mit der Automatisierung und Datenerhebung der Beete auf dem Dach der Hochschule Bochum sowie diese Daten zu Visualisieren.

2.4. Abkürzungen

MQTT: Message Queuing Telemetry Transport (offenes Netzwerkprotokoll für Machine-to-Machine-Kommunikation)

PV: Photovoltaik (eine Technologie zur direkten Umwandlung von Sonnenenergie in elektrische Energie)

MCU: Micro-Controller Unit

2.5. Teams und Schnittstellen

Rolle	Name	E-Mail
Teamleiter	Lucas Degenhardt-Seltmann	Lucas.degenhardt-seltmann@stud.hs-bochum.de
Mitglied	Florian Rüßler	florian.ruessler@stud.hs-bochum.de
Mitglied	Talha Taskaya	Talha.taskaya.@stud-hs.bochum.de
Mitglied	Tobias Maximilian Alexandre Basten	tobias.alexandre-basten@stud.hs-bochum.de
Betreuer	Stefan Jonker	Stefan.jonker@hs-bochum.de
Betreuer	Henrick Blunck	Henrik.blunck@hs-bochum.de
Betreuer	Marc-Maurice Torchala	marc.torchala@hs-bochum.de
Betreuer des OnTop Projektes	Christian Kazner	Christian.Kazner@hs-bochum.de
Gruppen Manager	Christian Kopse	ckopse10@gmail.com

3. Konzept

3.1. Ziel(e) des Anbieters

/ZANB10/ Wartungsfreie Fernsteuerung der Bewässerung und ausführliche Sammlung der Messwerte zur Auswertung.

/ZANB20/ Die Messgeräte müssen gut kalibrierbar und zuverlässig über längere Zeit autark funktionieren.

/ZANB30/ Der Server muss stabil und sicher laufen und mögliche Datenmengen verwalten können.

3.2. Ziel(e) und Nutzen des Anwenders

/ZANW10/ Automatisierte Bewässerung der Rooftop Farm Beete.

/ZANW20/ Erfasste Sensor Daten sowie Bewässerungs- und Photovoltaik Informationen auf einer Web-Oberfläche einsehen und abrufen können.

/ZANW30/ Eine Informative Web-Oberfläche für Projekt-fremde-Personen auf denen man Bilder, Live-Stream und interessant aufbereitete Daten einsehen kann.

3.3. Zielgruppe(n)

Sowohl die automatische Bewässerung als auch die erhobenen Sensordaten sind als ein großer Schritt für eine einfachere Verwaltung und Weiterführung aller Teams des Gesamtprojekts OnTop anzusehen. Die gewonnenen Statistiken können in Zukunft als handfeste Daten für Außenauftritte verwendet werden. Zudem ermöglicht das Projekt externen Interessenten, durch übersichtliche Einsicht der Livedaten, jederzeit einen deutlich tieferen Einblick in das Gesamtprojekt.

4. Funktionale Anforderungen

/F10/ Server mit Ubuntu als Betriebssystem und den Zusätzlichen Funktionen als Datenbank, MQTT, PHP-Host und einem zusätzlichem Massenspeicher für Livebilder. Außerdem sollen anhand der Auswertungen der Sensoren der Füllstand und Verbrauch der Wassertanks berechnet werden.

/F20/ Folgende MCUs mit Sensoren müssen an den Beeten installiert sein und entsprechende Daten an die Datenbank senden:

1x NodeMcu mit Hygrometer: Temperatur, Luftfeuchtigkeit

2x NodeMcu mit Ultraschall Distanz Sensor: Füllstand

3x NodeMcu mit Hygrometer und Ventil: Temperatur, Luftfeuchtigkeit;

/F30/ Außerdem muss das Ventil durch ein entsprechendes Signal vom Server zum Bewässern geschaltet werden.

/F40/ 1x Raspberry Pi mit Kamera, USB-PV Auswertung, Ultraschall Distanz Sensor: PV Auswertung, Füllstand; Außerdem werden Livebilder der Kamera in den Massenspeicher geladen

/F50/ Eine Wetterstation muss grundlegende Daten messen, die sich bei den Beeten nicht ändern würden und an die Datenbank senden.

/F60/ Auf einer Webseite und einer eigens entwickelten App können Externe Rechner und Smartphones die Sensordaten, Livebilder und den Wasserverbrauch einsehen. Diese sollen keine Rechte auf die Datenbank haben, sondern nur auf die Auswertungen.

5. Nichtfunktionale Anforderungen

5.1. Allgemeine Anforderungen

/NFA10/ Die Systeme und Programme müssen einfach verständlich und dokumentiert sein, damit diese leicht erweiterbar sind und von neuen Mitgliedern problemlos aufgenommen werden können.

/NF20/ Die Systeme müssen das Einfügen von neuen Sensoren und Beeten ermöglichen.

5.2. Gesetzliche Anforderungen

/NFG10/ Die üblichen Pflichten eines Website Betreibers an etwa den Datenschutz müssen eingehalten werden.

/NFG20/ Weil mit Elektronik gearbeitet wird müssen sowohl die DIN-VDE Normen erfüllt werden als auch alle Anforderungen an den Brandschutz.

5.3. Technische Anforderungen

/NFT10/ Reaktionszeit ist nicht wichtig, es reicht eine Minute später zu reagieren. Nur die historischen Daten zur Auswertung müssen schnell abgerufen werden können.

/NFT20/ Alle Systeme müssen pausenlos 24/7 laufen und erreichbar sein ohne aktiv betreut oder gesteuert zu werden. Ein kurzzeitiger Ausfall soll keine Probleme verursachen oder eine Wartung zur Inbetriebnahme benötigen.

6. Rahmenbedingungen

/R10/ Das Softwareprojekt ist über einen Zeitraum von zwei Semestern umzusetzen. Team intern wurden grobe Milestones erarbeitet, die weiter unten folgen werden.

/R20 Des Weiteren sind einige Komponenten vorhanden. Die restlichen Komponenten müssen nachgekauft werden. Nachdem alles Nötige vorhanden und funktionsfähig installiert ist, findet die Programmierung der einzelnen Komponente, zur Erfüllung des Projektplans, statt.

/R30 Schließlich muss das Projekt zur Benotung dokumentiert sein und andere Bedingungen wie z.B. dieses Pflichtenheft erfüllen.

6.1. Zeitplan

/Z10/ Sachen bestellen (2 Wochen) bis ungefähr Anfang Mai 2021

/Z20/ Module austesten (3 Wochen) bis ungefähr Ende Mai 2021

/Z30/ Module an die Beete (2 Wochen) anbringen bis ungefähr Mitte Juni

/Z40/ Server Aufsetzung und verbinden (3 Wochen) bis ungefähr Anfang Juli

/Z50/ Daten per MQTT an die Datenbank senden (4 Wochen) bis ungefähr Ende Juli

/Z60/ Webcam einrichten (3 Wochen) im nächsten Semester

/Z70/ Backend aufsetzen (3 Wochen) im nächsten Semester

/Z80/ Frontend aufsetzen (2 Wochen) im nächsten Semester

/Z90/ App fürs Handy (3 Wochen) im nächsten Semester

6.2. Technische Anforderungen

Benötigte Hardware:

- /TAH10/ Nodemcu
 - Ansteuerung und auslesen der Sensoren sowie der 12V Ventile, um die Beete mit Wasser zu versorgen
- /TAH20/ RaspberryPi 4
 - Füllstandsmessung der IB Container welche die zentrale Wasserversorgung stellen. Auslesen der Photovoltaik Anlage. Ansteuerung der Zentralen 12V Pumpe. Bereitstellung eines Kamera Feeds.
- /TAH30/ Wetterstation
 - Luft, Regen, Temperatur, Luftfeuchtigkeit messen
- /TAH40/ 12V Ventile
 - Ventil, um die Bewässerung der Beete steuern zu können.
- /TAH50/ Hygrometer
 - Sensor, um die Bodenfeuchte zu messen.
- /TAH60/ Distanz Sensor
 - Sensor, um den Füllstand der Wassercontainer zu messen.
- /TAH70/ 5V Schaltrelais
 - Schaltrelais, um mit einem 5V Strom den 12V Strom der Ventile schalten zu können.
- /TAH80/ VE.Direct zu USB-Interface
 - Interface, um die Daten der Photovoltaik Anlage über USB abrufen zu können.
- /TAH90/ Step Down Converter 12V zu 5V
 - Step Down Module, um die Nodemcus mit 5V Strom beliefern zu können.
- /TAH100/ Aufputz Zweigdosen
 - Wasserdichte Gehäuse für Nodemcu Boards.

Benötigte Software:

- /TAS10/ Mosquitto
 - Mqtt Broker als Kommunikationsendpunkt in dem sich alle Module einklinken.
- /TAS20/ InfluxDB
 - Database zur Sammlung der Daten.
- /TAS30/ Grafana
 - Frontend zur Visualisierung der Daten.
- /TAS40/ Nginx
 - Webserver der mit dem Media-Streaming-Module und Php-fpm ausgestattet sein muss, um die Basis für die Website liefern zu können.
- /TAS50/ Tasmota
 - Open Source Firmware für die NodeMCUs.

6.3. Problemanalyse

/P10/ Komplexität des Servers

/P20/ Ungenauigkeit der Daten benötigt eine angepasste Kalibrierung die entweder strenger oder lockerer sind. Ausreißer in den Daten müssen ignoriert werden!

/P30/ Die Stromversorgung könnte bei längerer Zeit ohne Sonne einbrechen, für diesen Fall sollte eine Gewichtung der Komponenten geschehen durch die unwichtigere abgeschaltet werden könnten, sowie auf eine Energieorientierte Programmierung geachtet werden.

/P40/ Der Energiebedarf darf nicht den Ertrag durch die Insel Photovoltaik Anlage übersteigen.

/P50/ Sensoren könnten nicht mit der freien Tasmota Firmware kompatibel sein so dass wir noch eigene Programme zur Inbetriebnahme der Sensoren schreiben müssen.

/P60/ Es könnte auch passieren das die Sensoren sich im Laufe des Projektes als nicht für den vorhergesehenen Zweck geeignet bzw. unzuverlässig herausstellen und diese somit ersetzt werden müssen.

/P70/ Die Wlan Verbindung könnte bei zu vielen Geräten Instabil werden, in diesem Fall müsste noch ein eigener Router für das Projekt aufgestellt werden.

6.4. Qualität

/Q10/ Die Daten müssen nützlich sein, um den Pflanzen automatisch, sparsam und genug Wasser zu geben und sie nicht zu überwässern. Dieses Kriterium wird durch Beobachtung des Pflanzenwachstums kontrolliert.

/Q20/ Die Protokollierung der Daten muss strukturiert sein, um gut ausgewertet werden zu können.

/Q30/ Eine hohe Energieeffizienz zu Sonnenarmen Zeiten ist sehr wichtig da ansonsten der Batteriespeicher nicht ausreichend sein könnte, dies wird durch Lastkurven und vorherige Bedarfsrechnungen gesichert.

/Q40/ Stabilität ist nicht zu vernachlässigen allerdings ist auch ein Ausfall von 1-2 Tagen nicht Systemkritisch.

7. Liefer- und Abnahmebedingungen

/A10/ Stefan Jonker und Henrick Blunck bewerten final das Projekt. Bei der Abgabe werden alle vorher genannten Schritte und Anforderungen integriert und lauffähig abgegeben.

/A20/ Weiterhin ist darauf zu achten, dass die Arbeit dieses Projektes empfänglich für zukünftigen Ausbau ist. Sie wird so gestaltet, dass Neuankömmlinge mit der Arbeit fortfahren können.

8. Anhang

Informationen zum kompletten Projekt:

<https://www.hochschule-bochum.de/studium-lehre/im-studium/studentische-projekte/on-top/>

GitHub Repository des Informatik Teilprojektes:

<https://github.com/LucasDegenhardtSeltmann/OnTopAutomatisierung>