

## Lastenheft

<Dennis Reinholz>  
<Automatische Balkonbewässerung>

Autor: Zul Vagala, Dzaid Siregar, Johannes Berg  
Letzte Änderung: 24.04.2025  
Dateiname: Automatische\_Balkonbewässerung\_Lastenheft  
Version: 3

## Inhaltsverzeichnis

1	Einleitung.....	4
2	Ausgangssituation.....	4
3	Zielsetzung.....	4
4	Anforderungen.....	5
4.1	Funktionale Anforderungen.....	5
4.2	Konstruktive Anforderungen.....	5
4.3	Software-Anforderungen.....	5
4.4	Technische Anforderungen.....	6
4.5	Nicht-Funktionale Anforderungen.....	6
4.6	Angestrebte Lösungsskizze.....	7
5	Abnahmekriterien.....	8
6	Ansprechpartner für Rückfragen.....	8
7	Wer hat was gemacht.....	8

## Copyright

© Mohammad Abuosba

Die Weitergabe, Vervielfältigung oder anderweitige Nutzung dieses Dokumentes oder Teile davon ist unabhängig vom Zweck oder in welcher Form untersagt, es sei denn, die Rechteinhaber/In hat ihre ausdrückliche schriftliche Genehmigung erteilt.

## Version Historie

Version:	Datum:	Verantwortlich
1	14.04.2025	Dennis
2	16.04.2025	Dzaid & Zul
3	24.10.2015	Dennis & Zul

## **1 Einleitung**

Pflanzen benötigen regelmäßige Pflege, insbesondere eine gleichmäßige und bedarfsgerechte Bewässerung. Viele Menschen haben jedoch im Alltag nicht immer die Zeit oder das nötige Wissen, um sich optimal um ihre Pflanzen zu kümmern. Genau hier setzt dieses Projekt an. Im Rahmen dieses Projekts wird ein automatisches Bewässerungssystem entwickelt, das mithilfe von Sensoren, einem Mikrocontroller und einer mobilen App arbeitet. Ziel ist es, eine intelligente Lösung zu schaffen, die den Feuchtigkeitsgehalt der Erde misst und die Pflanze bei Bedarf automatisch bewässert. Die Steuerung und Überwachung erfolgt bequem über das Smartphone.

## **2 Ausgangssituation**

Zurzeit erfolgt die Bewässerung von Pflanzen in der Regel manuell. Dabei ist es oft schwierig, die optimale Wassermenge und den richtigen Zeitpunkt genau einzuhalten – besonders bei wechselnden Umweltbedingungen wie Temperatur oder Luftfeuchtigkeit. Ohne technische Unterstützung erfordert dies regelmäßige Aufmerksamkeit und kann im Alltag schnell vergessen werden. Ein automatisiertes System könnte hier für Entlastung sorgen und die Pflanzenpflege effizienter gestalten.

## **3 Zielsetzung**

Das Ziel des Projekts ist es, ein kostengünstiges und einfach zu bedienendes System zu entwickeln, das die Pflanzenpflege automatisiert. Der Auftraggeber möchte eine Lösung, die zuverlässig erkennt, wann eine Pflanze Wasser benötigt, und diese dann automatisch bewässert. Dabei soll eine App dem Nutzer Informationen wie Bodenfeuchtigkeit, Temperatur und den aktuellen Zustand der Pflanze anzeigen. Langfristig soll das System den Alltag erleichtern und gleichzeitig einen Beitrag zur Nachhaltigkeit leisten, indem Wasser effizient eingesetzt wird.

## 4 Anforderungen

### 4.1 Funktionale Anforderungen

### 4.2 Konstruktive Anforderungen

Tabelle 1: Konstruktive Anforderungen

Nr.	Gruppe	Beschreibung	Priorität
1.	Feuchtigkeitssensoren	Das System muss kapazitive Bodenfeuchtesensoren verwenden, um den Feuchtigkeitsgehalt des Bodens präzise zu messen	Hoch
2.	Mikrocontroller	Ein ESP32-Mikrocontroller soll als zentrale Steuereinheit dienen, um Sensordaten zu verarbeiten und die Pumpe zu steuern	Hoch
3.	Bewässerungseinheit	Eine elektrisch gesteuerte Pumpe, die über ein Relaismodul vom ESP32 aktiviert wird, soll die Bewässerung durchführen.	Hoch
4.	Stromversorgung	Das System muss über eine stabile Stromquelle verfügen, entweder durch Netzstrom oder einen ausreichend dimensionierten Akku mit Solaranlage.	Hoch
5.	Network-Gateway	Das Gateway dient als Kommunikationsgerät zwischen dem ESP32 und das Internet	Mittel

### 4.3 Software-Anforderungen

Tabelle 2: Software-Anforderungen

Nr.	Beschreibung	Priorität
1.	Die App zeigt den aktuellen Status der Bewässerung in Echtzeit an.	Hoch
2.	Die Nutzer können die Bewässerung manuell über die App starten oder stoppen.	Hoch
3.	Die App soll automatisch Bewässerungszyklen basierend auf Sensorwerten steuern.	Hoch
4.	Fehler oder Warnungen sollen dem Nutzer per Push-Nachricht mitgeteilt werden.	Hoch
5.	Messdaten (z. B. Feuchtigkeit) sollen in der App in einer List dargestellt werden.	Mittel
6.	Die Software soll CI/CD-Pipelines für automatisierte Builds und Deployments nutzen.	Mittel
7.	Die Kommunikation zwischen App und Microcontroller erfolgt über MQTT.	Niedrig
8.	Die App soll mehrere Benutzerprofile unterstützen können.	Niedrig

## 4.4 Technische Anforderungen

Tabelle 3: Technische Anforderungen

Nr.	Gruppe	Beschreibung	Priorität
1.	Netzwerkverbindung	Der Mikrocontroller muss sich per WLAN mit einem lokalen Netzwerk oder Gateway verbinden können.	
2.	MQTT-Kommunikation	Das System muss das MQTT-Protokoll zur Datenübertragung verwenden. Der Mikrocontroller (ESP32) fungiert als Publisher, die mobile Anwendung als Subscriber. Die MQTT-Kommunikation erfolgt über ein lokales oder cloudbasiertes Gateway.	
3.	Datenformat	Alle übertragenen Daten sollen im JSON-Format codiert sein, um eine einheitliche, leicht interpretierbare Datenstruktur sicherzustellen.	

## 4.5 Nicht-Funktionale Anforderungen

Tabelle 4: Nicht-Funktionale Anforderungen

Nr.	Qualitätskriterien	Beschreibung	Priorität
1.	Zuverlässigkeit		
1.1		System muss nach einem Strom- oder Netzausfall automatisch wieder den Betrieb aufnehmen können	hoch
1.2		Sensorfehler oder Ausfälle einzelner Komponente sollen erkannt und gemeldet werden.	hoch
2.	Benutzbarkeit		
2.1		Die App soll eine intuitive Benutzeroberfläche mit hoch klaren Symbolen und verständlichen Anweisungen Bieten.	hoch
2.2		Bedienung soll mit wenigen Klicks möglich sein.	hoch
3.	Effizienz		
3.1		Das System soll energieeffizient arbeiten	mittel
3.2		Datenübertragung zwischen Controller und App sollen auf ein Minimum beschränkt werden, um Ressourcen zu sparen	niedrig
4.	Wartungsfreundlichkeit		
4.1		Der Quellcode muss modular und nach Clean-Code Prinzipien aufgebaut sein, um Wartung und Erweiterungen zu erleichtern.	hoch
4.2		Software soll durch automatische Tests überprüfbar sein.	mittel
5.	Übertragbarkeit		
5.1		System soll leicht an verschiedene Pflanzenarten und Balkonkonfigurationen anpassbar sein.	niedrig
5.2		App soll auf IOS und Android Geräten lauffähig sein, ohne Änderungen am Quellcode.	hoch

#### 4.6 Angestrebte Lösungsskizze (Modell)

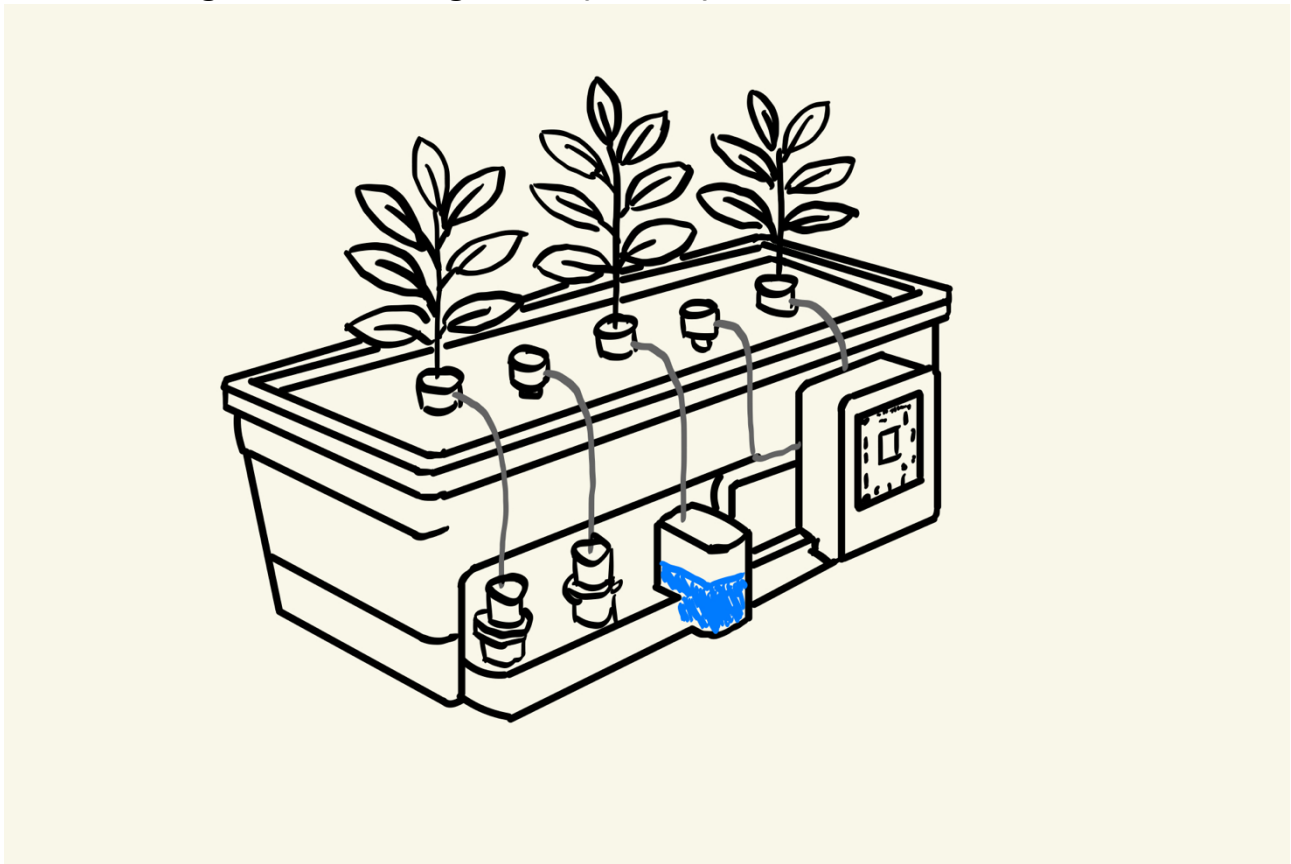


Abbildung 1: Modell

## 5 Abnahmekriterien

1. **Die App muss den aktuellen Status der Bewässerung in Echtzeit anzeigen.**  
*Das System gilt als abgenommen, wenn der Status der Bewässerung auf der Benutzeroberfläche automatisch aktualisiert wird.*
2. **Die manuelle Steuerung der Bewässerung über die App muss zuverlässig funktionieren.**  
*Die Funktion gilt als erfüllt, wenn der Nutzer die Bewässerung über einen Button in der App starten und stoppen kann.*
3. **Die automatische Bewässerung muss zuverlässig anhand der Sensorwerte gesteuert werden.**  
*Dieses Kriterium ist erfüllt, wenn bei Unterschreitung eines voreingestellten Feuchtigkeitsgrenzwerts die Bewässerung automatisch startet und nach Erreichen des Sollwerts wieder endet.*
4. **Fehlerzustände oder Warnungen müssen dem Nutzer aktiv durch eine Push-Benachrichtigung mitgeteilt werden.**  
*Das System wird akzeptiert, wenn es bei Ausfall eines Sensors oder bei leerem Wasserbehälter eine Benachrichtigung an den Nutzer innerhalb von 30 Sekunden versendet.*
5. **Die App muss auf iOS- und Android-Geräten mit einer intuitiven Benutzeroberfläche ausgestattet sein.**  
*Die App gilt als abnahmefähig, wenn alle Kernfunktionen auf beiden Plattformen fehlerfrei funktionieren und mindestens 80 % der Testnutzer die Bedienbarkeit als „intuitiv“ bewerten.*

## 6 Ansprechpartner für Rückfragen

Name	Mohammad Abuosba
Funktion	Projektleiter Auftraggeber
email	abuosba@htw-berlin.de
Telefon	+49 (30) 5019-3665

## 7 Wer hat was gemacht

Autor	Aufgabe/Kapitel	Anteil
Dzaid	Einleitung, Ausgangssituation, Zielsetzung	100%
	Funktionale & Technische Anforderungen	50%
Zul	Abnahmekriterien	100%
	Funktionale & Technische Anforderungen	50%
	Lösungsskizze (GUI)	50%
Johannes	Nicht-funktionale & konstruktive Anforderungen	100%
	Lösungsskizze (Modell)	50%