

The background of the slide is a close-up photograph of several large, green, wavy-edged leaves, likely from a plant like basil or spinach. The leaves are layered, with some in sharp focus and others blurred in the background, creating a sense of depth. The lighting is soft, highlighting the veins and textures of the leaves.

# Plant Health

plant today, enjoy tomorrow

703080: Software Engineering  
SS 2023

# Plant Health

plant today, enjoy tomorrow

Lehrveranstaltungsleiter:	Schweiger Harald Victor Mussmann Andrea Frankford Eduard Jäger Alexandra Simsek Umutcan Stricker Benedikt
Projektdauer	März, 2023 - Juni, 2023
Fakultät:	Informatik, Quality Engineering

Cover:	CC BY-NC-SA Harald Victor Schweiger
Style:	EPFL Report Style

# Unser Ziel

*Die fiktive NGO Plant-Health hat sich dem Ziel verschrieben die Menschheit wieder näher an die Natur zu bringen und Nahrungsmittelproduktion nicht nur auf den Feldern, sondern auch in privaten, öffentlichen und Büro-Gebäuden möglich zu machen. Auf diesem Weg sollen Probleme von Monokulturen und Lieferketten reduziert, Gebäude begrünt und die vor Ort anwesenden Menschen mit frischer und vitaminreicher Nahrung versorgt werden.*

*Um diesem Zielen näher zu kommen muss altes Wissen mit neuen Technologien verbunden werden, um souveräne Anbaulösungen auf eine neue Art und Weise anbieten zu können.*

# Produktidee

Pflanzen sind grundsätzlich souveräne Lebensformen. Drei Herausforderungen hindern jedoch die meisten Menschen daran Gemüse, Kräuter und mehr selber anzubauen.

1. **Zeitmangel:** Pflanzen müssen stetig überwacht werden. Sollte jedoch die Pflanze zu wenig Nährstoffe, Wasser oder Sonne bekommen oder erkranken, so müssen diese Probleme schnellstmöglich behoben werden, um eine erfolgreiche Ernte zu gewährleisten.
2. **Wissen:** Es braucht ab und zu ein geschultes Auge, um Probleme zu erkennen und beheben zu können.
3. **Umgebung:** In Innenbereichen sind konventionelle Anbaumethoden in Töpfen oft schwierig umzusetzen. Licht, Wärme und Feuchtigkeit sind hier, zum Teil baulich bedingt, vermindert zur Verfügung. Um eine erfolgreiche Ernte zu ermöglichen, ist genaue Überwachung der Pflanzengesundheit notwendig. Gleichzeitig wird bei konventionellen Lösungen oft der vorhandene Platz nicht vollständig ausgenutzt.

Plant-Health hat sich daher zum Ziel gesetzt, leere Flächen in privaten, öffentlichen und Büro-Gebäuden zum Anbau von Nutzpflanzen in Minigewächshäusern zu verwenden. Es soll eine technische Gesamtlösung entwickelt werden, die die Pflanzen direkt im Topf überwacht und relevante Informationen zu Umgebung und Pflanzengesundheit in einer geeigneten Softwarelösung benutzerfreundlich aufbereitet. Eigens angestellte Gärtner, aber auch Private sollen Zugang zur Softwarelösung haben, um die Pflanzen geeignet zu pflegen bzw., zu überwachen.

Es liegt an Ihrem Team ein Gesamtsystem in Büroräumen für den ersten Testlauf für Alpha-Nutzer zu entwickeln. Jedes Minigewächshaus ist mit den notwendigsten Erde-, Luft- und Lichtsensoren ausgestattet. Die Sensorwerte werden von einer Webanwendung gesammelt, gespeichert und für verschiedene Nutzergruppen entsprechend aufbereitet.

Bei Erfolg wird Plant-Health später eine zweite Version des Systems produziert, welche anhand von den gesammelten Daten und Machine Learning frühzeitig Mangelerscheinung und Krankheiten bei Pflanzen erkennen und Vorschläge zum Beheben geben wird.

# Inhaltsverzeichnis

<b>Unser Ziel</b>	<b>i</b>
<b>Produktidee</b>	<b>ii</b>
<b>1 Projektbeschreibung</b>	<b>1</b>
1.1 Sensoren und Mikrocontroller . . . . .	2
1.2 Accesspoint . . . . .	3
1.3 Webserver und Webapp . . . . .	3
1.4 Sicherheit . . . . .	4
1.5 Benutzerfreundlichkeit . . . . .	4
1.6 Logging . . . . .	5
1.7 Ausfallsicherheit . . . . .	5
<b>2 Weitere Vorgaben</b>	<b>6</b>

# Projektbeschreibung

Um den Zielen der NGO einen Schritt näher zu kommen, sollen Sie die erste Version einer Gesamtlösung zum Anbau von Kleinpflanzen in Miniaturgewächshäusern entwickeln. Diese Miniaturgewächshäuser sind dazu geeignet, in den Räumlichkeiten von Büro- und öffentlichen Gebäuden aufgestellt zu werden. Jedes Gewächshaus wird mit Sensoren ausgestattet, die Aufschluss über Erdfeuchtigkeit, Luftqualität, Lichtintensität und mehr geben. Jeder Bürotrakt, in dem Miniaturgewächshäuser stehen, ist mit einem Accesspoint ausgestattet. Der Accesspoint überprüft laufend, ob die für die Sensordaten vorgegebenen Grenzwerte im gesetzten Rahmen sind und initiiert bei über- oder unterschreiten, visuelle Warnungen an der betreffenden Sensorstation. Die Accesspoints senden periodisch die neuesten Daten an den Webserver wo sie permanent gespeichert und für die User entsprechend ihrer Berechtigungen aufbereitet werden.

Kernstück des Projektes ist eine Multi-User-Webanwendung, die die gesammelten Messwerte der Sensoren auswertet und benutzerfreundlich in geeigneten Ansichten aufbereitet. In einem von Ihrem Team entworfenen Dashboard sehen all jene Anwender mit entsprechenden Berechtigungen auf einem Blick, ob in den Miniaturgewächshäusern alles in Ordnung ist, oder ob einzelne Messwerte über- bzw. unterschritten wurden und somit Handlungsbedarf an einer Pflanze besteht. Zusätzlich können Bilder der unterschiedlichen Pflanzen zu verschiedenen Aufnahmezeitpunkten angezeigt werden. Administratoren können am Dashboard außerdem sehen, ob alle Sensorstationen und Accesspoints erreichbar sind.

Gärtner haben die Möglichkeit, sich historische Messdaten der ihnen zugeordneten Gewächshäuser in geeigneten Graphen und frei wählbaren Zeiträumen anzeigen zu lassen. Damit ist den Gärtnern die Möglichkeit gegeben, Veränderungen an der Pflanze z.B. am Gießverhalten, Standort, oder jahreszeitlich bedingtem Lichteinfall durch Trends in den Messdaten nachzuvollziehen und langfristig optimale Standorte für die unterschiedlichen Nutzpflanzen zu identifizieren.

Die Miniaturgewächshäuser sind außerdem mit einem jeweils eindeutigen QR-Code versehen. Alle die diesen QR-Code mit dem Smartphone einlesen werden auf eine Website in der Webapp weitergeleitet, auf der ohne Login Bilder der betreffenden Pflanze angesehen und gegebenenfalls neue hochgeladen werden können.

Behälter, Erde, Samen, Sensoren und andere Hardware werden Ihrem Team von der NGO zur Verfügung gestellt. Der Behälter ist wiederverschließbar und daher für den Transport zwischen verschiedenen Orten bestens geeignet. Eine Darstellung zum Aufbau des Projektes finden Sie in Abbildung 1.1.

In den nachfolgenden Abschnitten werden weitere Details und Vorgaben zum Projektaufbau genauer beschrieben. Die Gliederung ist hierbei weitgehend nach den Hardwarekomponenten strukturiert.

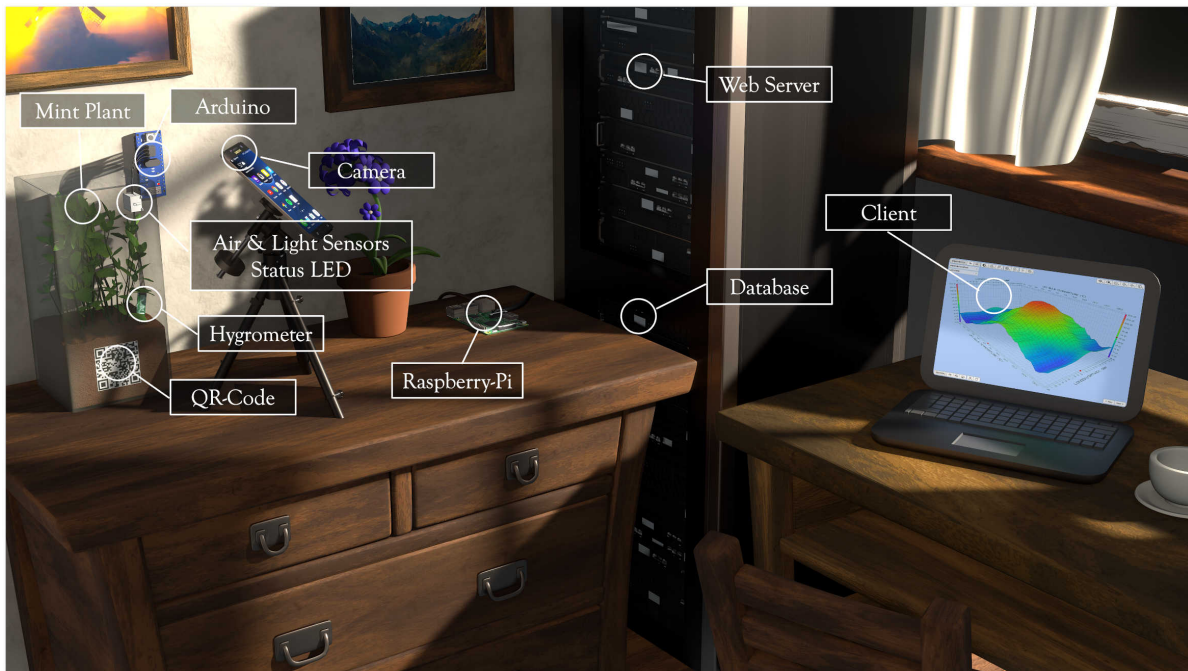


Abbildung 1.1: Die gesamte Infrastruktur des Projektes

## 1.1. Sensoren und Mikrocontroller

Die Basisausstattung an Sensoren der Produktpalette von Plant-Health mit der jedes Miniaturgewächshaus überwacht wird, wird Ihrem Team von Plant-Health zu Verfügung gestellt. Das Produkt umfasst dabei die folgenden Sensoren:

- Einen **Erdfeuchtigkeitssensor** (Hygrometer) , der den Wassergehalt in der Pflanzenerde bestimmt.
- Einen **Luftsensord** (BME688) , der Luftfeuchtigkeit, Luftdruck, Temperatur und Luftqualität misst. Es sei hierbei angemerkt, dass die Luftqualität mittels eines VOC-Sensors, welcher die Menge an flüchtigen organischen Verbindungen erkennt, gemessen wird. Welcher Wert für welche Pflanze optimal ist, ist bisher nicht ausreichend erforscht und soll daher im Zusammenhang mit den anderen Werten aufgezeichnet werden. Ein zu hoher Wert kann aber im schlimmsten Fall auf einen Schimmelbefall, meistens ausgelöst durch falsches Gießen, hindeuten.
- Einen **Lichtsensord** (Fototransistor) , der die Lichtintensität misst. Es muss hierbei berücksichtigt werden, dass der Sensor in der von Ihnen umzusetzenden Iteration nicht zwischen künstlichem Licht und Sonnenlicht unterscheiden kann.

Zusätzlich ist jedes der Miniaturgewächshäuser mit einer LED-Lampe und einem Lautsprecher ausgestattet, welche visuelle und akustische Informationen über den Verbindungsstatus, bzw. ob die Messwerte innerhalb der konfigurierten Intervalle liegen, geben soll. Genauere Details über die Konfiguration der Messwerte siehe auch Sektion 1.2 und Sektion 1.3. Weitere Informationen zu den visuellen und akustischen Hinweisen, siehe Abschnitt 1.5.

Die Sensoren des Miniaturgewächshauses sind mit einem Mikrocontroller (Arduino) verbunden. Ein externes Team wird nach der Projektfertigstellung damit beauftragt den Arduino in die finale Mikrocontroller Version umzuwandeln. Während der initialen Testphase darf der Arduino über ein Kabel mit Strom versorgt werden. Die spätere Version des Produkts wird jedoch energieeffizienter und somit über Batterie und mittels *wireless charging* betrieben. Daher ist es wichtig schon jetzt die Sensordaten stromsparend zu übertragen. Hierbei soll Bluetooth-Low-Energie (BLE) verwendet werden, um Messdaten an eine mit dem Internet verbundenen Station (Wireless Accesspoint) zur übertragen.

Für die Entwicklung des Accesspoint wird Ihnen hierbei ein *Raspberry-Pi 4B* , der mit *WLAN 802.11ac* und *Bluetooth 5.0* ausgestattet ist, zur Verfügung gestellt. Die Sensorstation in Kombination mit dem Accesspoint bilden einen Cluster innerhalb eines Bürotraktes. Innerhalb des Clusters besitzt jede Sensorstation eine eindeutige Identifikationsnummer, die mittels eines an der Sensorstation befestigten

*DIP-Schalter*s definiert werden kann. Die Stationen speichern immer die aktuellsten Werte der Sensoren in einem flüchtigen Speicher (Volatile Memory). Da gewisse Sensoren zum Rauschen (Varianz in den Messungen) neigen, soll darauf geachtet werden, dass die Daten zwischen den Übertragungsintervallen akkumuliert werden.

Es ist wichtig, dass die Daten nur an einen autorisierten Accesspoint gesendet werden. Hierfür gibt es an der Sensorstation einen Taster, der den Mikrocontroller in einen Kopplungsmodus versetzt. Der erste Accesspoint, der sich in diesem Zeitraum mit der Sensorstation verbindet, ist berechtigt, Daten abzufragen. Der Kopplungsmodus soll dabei nicht länger als fünf Minuten anhalten und soll vorzeitig abgeschlossen werden, wenn sich ein Accesspoint erfolgreich verbinden konnte.

## 1.2. Accesspoint

Da es sich beim Accesspoint vorerst um einen Raspberry-Pi handelt erwartet Plant-Health, dass Sie das Betriebssystem RaspberryPi-OS verwenden und die benötigte Software des Accesspoints entweder in Python3 oder in Java entwickeln. Plant-Health hat Ihnen dazu Mitgeteilt, dass nach ersten Recherchen die Unterstützung von Bluetooth LE in Python wesentlich besser ist als in Java.

Auf dem Accesspoint befindet sich eine Datenbank, die die Daten der Sensorstationen zeitlich begrenzt persistent speichert. Diese Daten werden in einem konfigurierbaren Intervall an den Webserver gesendet und bei erfolgreicher Übermittlung aus der Datenbank gelöscht.

Auf dem Accesspoint sind außerdem für jeden Sensor an jedem verbundenen Gewächshaus die Grenzwerte hinterlegt, die die Messwerte nicht unter- oder überschreiten sollen. Wird bei den Daten, die sich der Accesspoint von einer Sensorstation holt, eine Grenzwertüberschreitung über eine längere (konfigurierbare) Dauer festgestellt, muss der Accesspoint die LED-Lampe auf der betroffenen Sensorstation ansteuern und einen entsprechenden Farb- und Blinkcode aktivieren. Weitere Informationen dazu finden Sie in Abschnitt 1.5.

Das Hinzufügen von neuen Sensorstationen zu einem Accesspoint wird über die Weboberfläche initiiert. Administratoren initiieren zuerst die Suche nach neuen Sensorstationen, die vom Accesspoint eindeutig als solche erkennbar sein müssen und weisen danach die Sensorstationen einem Accesspoint zu. Für die zugewiesenen Sensorstationen wird am Webserver ein Datenobjekt erstellt und persisten in der Datenbank gespeichert. Es ist wichtig, dass der Accesspoint nicht eigenständig auf die Suche nach neuen Stationen geht.

Es gibt somit drei Anlässe, warum der Accesspoint mit einer Sensorstation kommuniziert:

1. Es sollen Sensordaten an den Accesspoint gesendet werden.
2. Beim Übermitteln der Sensordaten wurde eine langfristige Grenzwertüberschreitung festgestellt, die der Sensorstation mitgeteilt werden soll.
3. Die Sensorstation befindet sich im Kopplungsmodus und wartet darauf, sich erfolgreich mit einem Accesspoint zu verbinden.

Da sich der Accesspoint nur mit genau einem Webserver verbinden soll, wird die IP-Adresse des Webserver in eine Konfigurationsdatei mit dem Namen *conf.yaml* gespeichert. Damit sämtliche Einstellungen während der Installation der Accesspoints gesetzt werden können, soll ein Skript, welches mittels `./configure` ausgeführt wird, bereitgestellt werden. Zusammengefasst sollen die folgenden Einstellungen durch das Skript und der Konfigurationsdatei einstellbar sein:

- Das initiale Übertragungsintervall zwischen Accesspoint und Webserver.
- Die Adresse des Webserver.
- Einen Namen der sich auf den Gebäudetrakt oder Raum bezieht, in dem sich der Accesspoint befindet.

## 1.3. Webserver und Webapp

Wie in der Einführung beschrieben läuft am Webserver eine Webanwendung, die die gesammelten Messwerte der Sensoren auswertet und benutzerfreundlich in geeigneten Ansichten aufbereitet. Die Basis dieser Webanwendung ist ein Java Spring Boot Skeleton Projekt, dass Ihrem Team von Plant-Health zur Verfügung gestellt wird. Die Daten der Anwender und sämtliche Sensordaten sollen einheitlich und persistent in einer MySQL-Datenbank gespeichert werden.



Alle graphischen Auswertungen sollen übersichtlich gestaltet sein, Möglichkeiten zur Filterung bieten und auf benutzerdefinierte Zeiträume beschränkt werden können. Grenzwertverletzungen (sowohl Über-, als auch Unterschreitungen) sollen in allen Darstellungen gut sichtbar sein. Außerdem sollen berechnete Anwender die Möglichkeit haben, sich tabellarische Auflistungen der einzelnen Sensorwerte anzeigen zu lassen. Alle Auflistungen müssen ebenso filter- und sortierbar sein.

Plant-Health möchte in der Webapp folgende Rollen umgesetzt sehen:

- **Administratoren** können Nutzer hinzufügen, bearbeiten und entfernen. Sie besitzen zudem auch die Rechte aller anderen Nutzer. Weiters können Administratoren Gewächshäuser (bzw. deren Sensorstationen) editieren, löschen und jedem Gewächshaus einen Universally Unique Identifier (UUID) zuweisen. Die UUID kann als QR-Code ausgedruckt werden und soll an dem Gewächshaus befestigt werden. Administratoren können auch neue Accesspoints anlegen, editieren und löschen. Administratoren sind daher für die Kopplung zwischen Accesspoints und den Sensorstationen verantwortlich. Administratoren sind auch dafür verantwortlich die Gewächshäuser registrierten Gärtnern zuzuordnen.
- **Gärtner** können die Daten der ihnen zugeteilten Gewächshäuser einsehen und Einstellungen wie zum Beispiel Grenzwerte und Übertragungsintervalle zwischen Accesspoint und Webserver treffen. Gärtner verwalten auch die Bildergalerie ihrer Pflanzen.
- **User** können das Dashboard einsehen und auswählen, welche Pflanzen ihnen dort angezeigt werden.
- **Besucher** können mittels Smartphone die QR-Codes, die an den Gewächshäusern befestigt sind, einscannen. Sie werden hierbei an eine Webseite der Webapp weitergeleitet, auf der sie Fotos der jeweiligen Pflanze in eine eigens dafür vorgesehene Bildergalerie hochladen können. Diese Rolle erfordert keinen Login. In dieser ersten Iteration der Software müssen hier auch keine Kontrollmechanismen implementiert werden.

Die Messdaten aller Sensoren, die temporär in den Accesspoints hinterlegt sind, werden zur permanenten Speicherung auf den zentralen Webserver übertragen. Der Webserver muss eine geeignete REST Schnittstelle zu Verfügung stellen, über die sowohl Sensordaten, als auch Konfigurationseinstellungen von bzw. an die Accesspoints übertragen werden können.

Sowohl die Einstellungen für die Datenbankverbindungen, als auch für die Speicherung der Pflanzenbilder, die Besucher in die Bildergalerie hochladen können, muss konfigurierbar sein.

## 1.4. Sicherheit

Das Softwaresystem wird im Anschluss von den Sicherheitsexperten der NGO geeignet gesichert. Damit Sie aber schon während der Entwicklung sicher stellen können, dass Sie nur Daten von den von Ihnen aufgestellten Accesspoints erhalten, werden alle Accesspoints, die sich erfolgreich mit dem Webserver verbinden konnten, dem Administrator zum Konfigurieren vorgeschlagen und müssen manuell freigegeben werden. Eine Freigabe muss auch für jede Sensorstation erfolgen. Vergewissern Sie sich, dass nur dann Messdaten von Sensorstationen an den Webserver gesendet werden, wenn der Administrator sowohl die jeweilige Sensorstation als auch den Accesspoint entsprechend aktiviert/freigeschaltet hat. Andernfalls sollen Anfragen zur Übertragung der Daten frühestens von der Sensorstation und spätestens von dem Webserver abgelehnt werden.

## 1.5. Benutzerfreundlichkeit

Beim Einschalten und erfolgreicher oder fehlgeschlagener Registrierung der Sensorstation soll eine akustische Tonfolge abgespielt werden. Während der Verbindungsphase soll die LED der Sensorstation blau blinken.

Sollten die gesetzten Grenzwerte der Sensoren unter- oder überschritten werden, so soll sich die LED der Sensorstation ähnlich zu einem Farbverlauf von grün auf orange bis hin zu rot verfärben. Gleichzeitig soll jedem gemessenen Umgebungswert ein eigener Blinkcode zugeordnet sein. Aus der Kombination aus Farbe und Blinkcode sollen Anwender erkennen können, welche Messwerte problematisch sind. Diese Information muss für den Anwender auch aussagekräftig in der Webapp angezeigt werden. Die Warnungen bleiben solange aufrecht, bis sie ein Gärtner einen eigens für diesen Zweck montierten Taster auf der Sensorstation betätigt. Diese Information, dass ein Gärtner vor Ort ist und die Warnungen deaktiviert hat, muss in der Webapp geeignet angezeigt werden.

Alle Systemteile müssen mittels Installationsanleitung problemlos auf den jeweiligen Geräten einzu-richten sein. Für die Accesspoints wird zusätzlich ein Konfigurationsskript und ein Installationsskript verlangt.

## 1.6. Logging

Das System soll sowohl Log Files schreiben, als auch wichtige Ereignisse in ein Audit-Log speichern. Administratoren haben Zugang zum Audit-Log des Webserver. In dieses werden besondere Ereignisse gespeichert, die im Zuge eines Audits aus Gründen der Nachverfolgbarkeit zu Verfügung stehen müssen. Das Audit-Log soll über die Weboberfläche einsehbar und nach verschiedenen Ereignistypen durchsuchbar sein.

Außerdem werden alle wichtigen Ereignisse in zwei unterschiedlichen Log-Dateien direkt am Webserver gespeichert. Eine der Log-Dateien logt alle Informationen, die andere nur die Warnungen und Fehlermeldungen. Zu Ereignissen zählen zum Beispiel das Hinzufügen neuer Verbindungen oder Geräte, sämtliche CRUD-Operationen an den Benutzern und Ausfälle der Sensorstationen, Accesspoints oder des Webserver.

Ebenso sollen eine Log-Datei am Accesspoint geschrieben werden. Bitte loggen Sie hier die Kommunikation mit den Sensorstationen, Übertragungen an den Webserver, und Verbindungsprobleme bzw. -ausfälle sowohl mit Sensorstationen als auch dem Webserver.

Wichtige Informationen, die pro Ereignis gespeichert werden, umfassen mindestens das Datum, die Uhrzeit, den Benutzer, der das Ereignis ausgelöst hat, und Informationen zum Ereignis selbst.

Wo die Log Dateien gespeichert werden muss außerdem sowohl am Webserver, als auch am Raspberry konfigurierbar sein.

## 1.7. Ausfallsicherheit

Es kann weder von einer dauerhaft stabilen Stromversorgung ausgegangen, noch kann garantiert werden, dass die Kommunikation zwischen Sensorstation und Accesspoint, bzw. zwischen Accesspoint und Backend störungsfrei verläuft.

Folgende Ausfallsszenarien sollten Sie in der Entwicklung beachten und durch entsprechende Lösungen abdecken:

- T1: Eingeschränkte Kommunikation zwischen Arduino Sensorstationen und Minirechner
- T2: Unerwarteter Neustart eines Minirechners
- T3: Temporärer Ausfall der Kommunikationswege zwischen Minirechnern und zentralem Backend
- T4: Kurzfristiger Ausfall des zentralen Backends.
- T5: Ausfall der Sensorstation

Es ist wichtig, dass alle Accesspoints und Sensorstationen automatisch nach einem Stromausfall wieder hochfahren und ihre Dienste selbständig und ohne Zutun von Außen aufnehmen. Es soll außerdem im Besonderen darauf geachtet werden, dass die ACID Bedingungen für Datenbanktransaktionen eingehalten werden.

Ausfälle von Sensorstationen oder Accesspoints müssen in der Webapp und in den Log-Dateien einsehbar sein. Außerdem muss diese Information in die Auswertungen der Messwerte miteinfließen, um das Verfälschen der Auswertungen durch fehlende Datenpunkte zu vermeiden.

# 2

## Weitere Vorgaben

Folgende zusätzliche Anforderungen wurden Ihnen mitgeteilt:

- Webbasierte Multi-User Anwendung, lauffähig und bedienbar auf allen gängigen Plattformen (PC, Tablet, Smartphone) und aktuellen Internet-Browsern
- Zulässige Technologien (Webserver): Java 17, Spring, OmniFaces, PrimeFaces, MySQL (auf Basis des SWE Skeleton-Projekts<sup>1</sup>). Grundsätzlich steht es Ihnen frei, ein anderes UI Framework zu wählen. Sie müssen allerdings in der Lage sein, dieses selbstständig mit Spring und SonarQube zu konfigurieren.
- Verpflichtende Verwendung der Universitäts-GitLab Instanz<sup>2</sup> zur Verwaltung des Source Codes
- SonarQube soll verwendet werden, um regelmäßig die Qualität des Webapp Source Codes zu überprüfen und zu verbessern
- Zulässige Technologien (Arduino): C/C++. Es wird empfohlen den Code mit der Programmierumgebung VSCode<sup>3</sup> und PlatformIO<sup>4</sup> zu entwickeln.
- Design und Abgabe eines Anschlussplans für die Sensoren. Der Plan muss von der PS-Leitung genehmigt werden, bevor mit dem Bau der Sensorstation begonnen werden darf.
- Zulässige Technologien (Accesspoint): Python  $\geq 3.9$ , bluezero<sup>5</sup> oder bleak<sup>6</sup>. Anstelle von Python kann Java verwendet werden, wird aber nicht empfohlen.
- GATT Services und Charakteristiken müssen so weit wie möglich aus der Bluetooth Dokumentation (GATT Specification Supplement)<sup>7</sup> entnommen werden.
- Änderungen am Technologie-Stack dürfen nur in Absprache mit der Proseminarleitung getroffen werden. Dasselbe gilt für den Wechsel von Major Releases eingesetzter Technologien und Frameworks.
- Modulare Softwarearchitektur unter Einhaltung gebräuchlicher Qualitätsmerkmalen und korrekter Anwendung von Architektur- und Entwurfsmustern
- Anforderungen an die Software- und Codequalität Webapp: Testabdeckung von mindesten 65% (JUnit, Überprüfung durch SonarQube). Controller Klassen für UI-Funktionalität sollen aus der Berechnung der Testabdeckung ausgenommen werden.
- Anforderungen an die Software- und Codequalität Accesspoint: Hinreichende Testabdeckung (Unittests).
- Bereitstellung realitätsnaher Testdaten in ausreichender Qualität und Quantität
- Einheitliche Formatierung von Quellcode im gesamten Projekt

---

<sup>1</sup><https://git.uibk.ac.at/informatik/qe/swe-skeleton>

<sup>2</sup><https://git.uibk.ac.at>

<sup>3</sup>VSCode: <https://code.visualstudio.com/>

<sup>4</sup>PlatformIO: <https://platformio.org/>

<sup>5</sup>Bluezero Python Bluetooth Library: <https://github.com/ukBaz/python-bluezero>

<sup>6</sup>Bleak Python Bluetooth Library: <https://pypi.org/project/bleak/>

<sup>7</sup>Bluetooth Spezifikation: <https://www.bluetooth.com/de/specifications/assigned-numbers/>

- Angemessene Dokumentation des Quellcodes und eingesetzter Installations- und Konfigurationsfiles
- Angemessene Verzeichnis- und Projektstruktur
- Keine unerlaubte Nutzung urheberrechtlich geschützter Werke, keine Verletzung von Lizenzbestimmungen. Fremde Inhalte müssen klar als solche inkl. Quellenangabe gekennzeichnet werden.