

PS Softwareentwicklung und Projektmanagement

PS1-Team 2

# Konzeptbeschreibung

Cotrotzo Marco, Glaser Benedikt, Jasharoski Medin,  
Kirchmair Andreas, Müller Jona

# 1 Systemüberblick

Diese Software soll es ermöglichen, viele kleine, zu Clustern zusammengefasste, Gewächshäuser zentral zu verwalten.

Um dies zu bewerkstelligen, sind in jedem Gewächshaus auf einem Arduino mehrere Sensoren verbaut, die über Bluetooth LE Messdaten an einen AccessPoint schicken. Die Gewächshäuser sind batteriebetrieben. Im Falle einer Messwert Unter- bzw. Überschreitung kann diese über eine RGB-LED kommuniziert werden.

Ein AccessPoint kann sich mit bis zu 255 Gewächshäuser (SensorStations) verbinden. Die zugeschickten Daten werden auf Messwert Unter- bzw. Überschreitungen geprüft und diese werden der entsprechenden SensorStation ggf. mitgeteilt. Die Daten werden in einstellbaren Intervallen an einen zentralen Webserver übertragen, über dessen Weboberfläche verschiedene Akteure auf die Daten zugreifen und diese verwalten können.

Des Weiteren ist an jedem Gewächshaus ein QR-Code angebracht, über den man auf eine Seite gelangt, auf der man Bilder der Pflanze ohne Anmeldung hochladen kann.

## 2 Use-Cases

### 2.1 Akteure

**Admin:**

Ein Admin verwaltet die User, AccessPoints und SensorStations des Systems

**Gardener:**

Ein Gardener verändert Grenzwerte und Übertragungsintervall der ihm zugeteilten SensorStations. Des Weiteren kann er Fotos zu seinen Pflanzen löschen.

**User:**

Ein User kann Messdaten von beliebigen Pflanzen einsehen.

**Visitor:**

Ein Visitor kann an der SensorStation einen QR-Code scannen, über den er auf eine Seite geleitet wird, an der er Fotos der Pflanze in der Station hochladen oder betrachten kann.

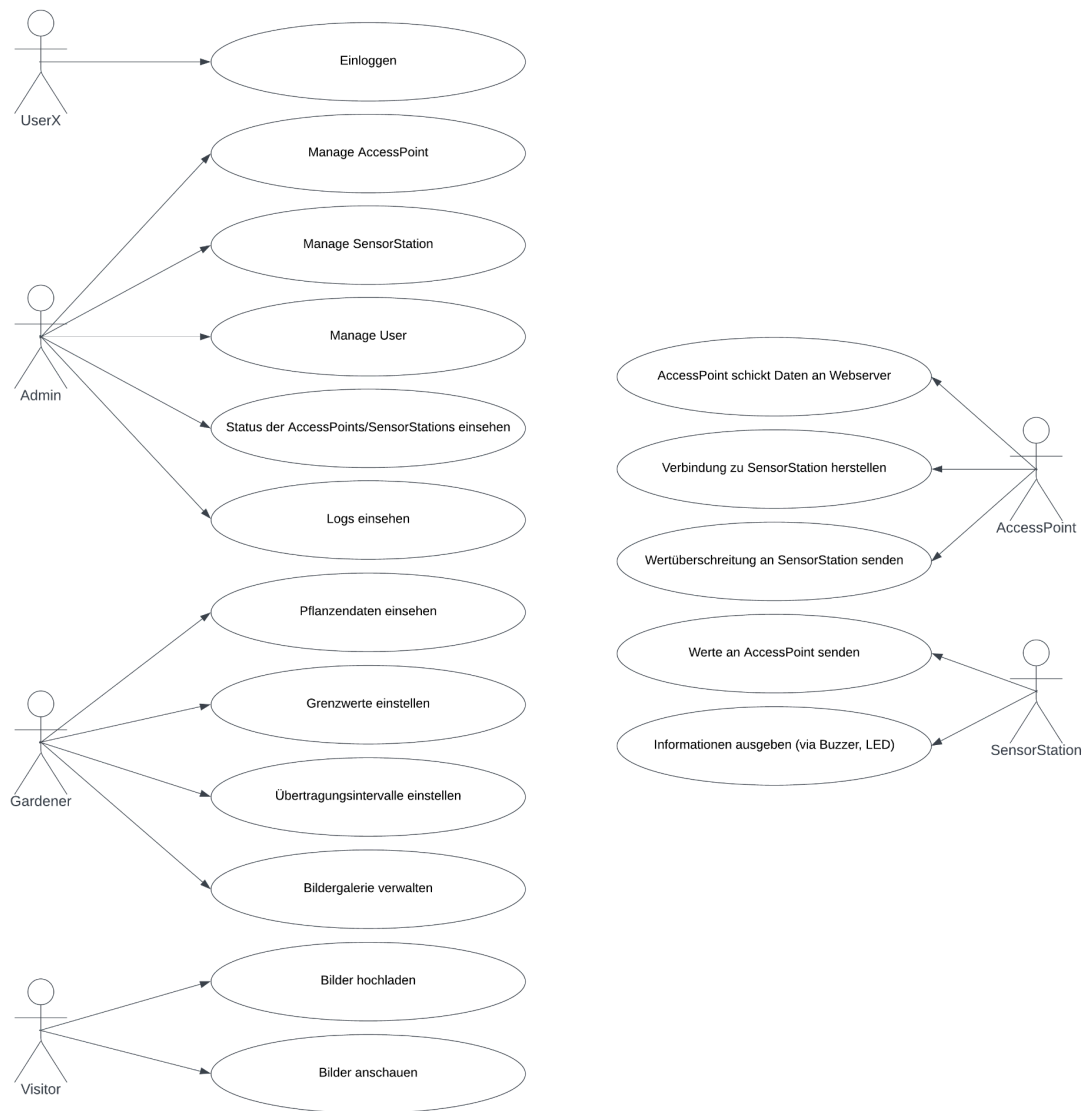
**AccessPoint:**

Ein Access empfängt Daten von bis zu 255 SensorStations und überprüft diese auf Grenzwerte Unter- bzw. Überschreitungen. Diese werden in einstellbaren Intervallen an den Webserver gesendet. Er kann sich im Pairing-Modus mit neuen SensorStations verbinden.

**SensorStation:**

Eine SensorStation beinhaltet eine Pflanze und mehrere Sensoren. Die Daten dieser Sensoren werden an einen AccessPoint gesendet. Im Falle, dass dieser einen Grenzwert Unter- bzw. Überschreitungen feststellt wird diese über eine RGB-LED ausgegeben.

## 2.2 Use-Case Diagramm



## 2.3 Use-Cases

### 2.3.1 Akteur: User

#### Einloggen:

- **Vorbedingung:** Das System läuft, man befindet sich auf der Startseite und der User soll von Admin schon erstellt worden sein.
- **Ablauf:** Benutzer trägt den Benutzernamen und das Passwort ein und wird durch das Drücken auf den Login-Button weitergeleitet.
- **Erfolg:**
  - Admin: Alle Gewächshäuser werden angezeigt.
  - Gärtner: Die Gewächshäuser, für die der angemeldete Gärtner zuständig ist, werden angezeigt.
  - Benutzer: Können das Dashboard mit allen Gewächshäusern einsehen.
- **Kein Erfolg:** Der User erhält eine Fehlermeldung.
- **Involvierte Klassen:** User.

### 2.3.2 Akteur: Gardener

#### Pflanzendaten einsehen:

- **Vorbedingung:** Das System läuft, der Nutzer ist mit Berechtigungslevel Gardener angemeldet.
- **Ablauf:** Der Nutzer wählt die Sensorstation mit der Pflanze, deren Daten er einsehen will, aus dem Dashboard aus und gelangt auf eine neue Seite.
- **Erfolg:** Auf dieser Seite sind u.a. die (historischen) Daten für diese Pflanze in grafischer und Textform aufgelistet.
- **Kein Erfolg:** Der Nutzer erhält eine Fehlermeldung.
- **Involvierte Klassen:** Gardener

#### Grenzwerte einstellen:

- **Vorbedingung:** Das System läuft, der Nutzer ist mit Berechtigungslevel Gardener angemeldet.
- **Ablauf:** Der Nutzer wählt die Sensorstation mit der Pflanze, deren Grenzwerte er ändern will, aus dem Dashboard aus und gelangt auf eine neue Seite. Auf dieser Seite werden u.a. die aktuellen Grenzwerte angezeigt.
- **Erfolg:** Die Grenzwerte können bearbeitet werden.
- **Kein Erfolg:** Der Nutzer erhält eine Fehlermeldung.
- **Involvierte Klassen:** Gardener

### Übertragungsintervalle einstellen:

- **Vorbedingung:** Das System läuft, der Nutzer ist mit Berechtigungslevel Gardener angemeldet.
- **Ablauf:** Der Nutzer wählt die Sensorstation, deren Übertragungsintervalle er ändern will, aus dem Dashboard aus und gelangt auf eine neue Seite. Auf dieser Seite wird u.a. das aktuelle Übertragungsintervall angezeigt.
- **Erfolg:** Das Übertragungsintervall kann bearbeitet werden.
- **Kein Erfolg:** Der Nutzer erhält eine Fehlermeldung.
- **Involvierte Klassen:** Gardener

### Bildgalerie verwalten:

- **Vorbedingung:** Der User ist angemeldet und hat die Rolle „Gardener“.
- **Ablauf:** Der Gärtner sieht alle seine Pflanzen aufgelistet. Er klickt auf einen Button einer bestimmten Pflanze, um die Bildergalerie zu öffnen. Der Gärtner hat nun die Möglichkeit, Bilder zu löschen, indem er auf einen Button am Bild klickt.
- **Erfolg:** Die Galerie erscheint und der Gärtner kann die Bilder verwalten.
- **Kein Erfolg:** -
- **Involvierte Klassen:** Userx, PlantPicture, SensorStation

## 2.3.3 Akteur: Admin

### Manage Access Point:

- **Vorbedingung:** Der User hat sich als Admin erfolgreich angemeldet.
- **Ablauf:**
  - Admin wechselt zu der Sicht der AccessPoints und klickt auf den Verwaltungs-Button
  - Folgendes kann der Admin nun machen:
    - Neues AccessPoint anlegen
    - Existierender AccessPoint bearbeiten
    - Existierender AccessPoint löschen
- **Erfolg:** Admin hat einen neuen AccessPoint angelegt, einen bereits vorhandenen bearbeitet oder einen gelöscht.
- **Kein Erfolg:** Fehlermeldung
- **Involvierte Klassen:** User mit Admin Berechtigungen, AccessPoint.

### Manage Sensor Station:

- **Vorbedingung:** Der User hat sich als Admin erfolgreich angemeldet.
- **Ablauf:**
  - Admin kann über den Header auf der Sicht der Sensor Stationen gelangen
  - Dort kann man dann bei der ausgewählten Sensor Station auf den Verwaltungs-Button klicken.
  - Folgendes kann der Admin nun machen:
    - Sensor Station löschen
    - Sensor Station editieren
    - Der Sensor Station einen Universally Unique Identifier (UUID) zuweisen?
- **Erfolg:** Der Sensor Station wird einen UUID zugewiesen, diese wird editiert oder gelöscht.
- **Kein Erfolg:** Fehlermeldung
- **Involvierte Klassen:** User mit Admin Berechtigungen, SensorStation.

### Verbindung zum Webserver herstellen:

- **Vorbedingung:** Webserver läuft, Admin eingeloggt, AccessPoint eingeschaltet
- **Ablauf:**
  - Name des AccessPoints und Adresse des Webserver werden am Accesspoint konfiguriert. AccessPoint versucht, sich mit dem Webserver zu verbinden.
  - Verbindung erfolgreich hergestellt. Admin sieht im Webinterface den neuverbundenen Accesspoint.
  - Admin gibt den AccessPoint manuell im Webinterface frei.
- **Erfolg:** AccessPoint ist mit dem Webserver verbunden und Kommunikation ist möglich.
- **Kein Erfolg:** Verbindung fehlgeschlagen. Erneuter Verbindungsversuch.
- **Involvierte Klassen:** Admin, AccessPoint

## Manage User:

- **Vorbedingung:**  
Das System läuft, man ist auf der Startseite und als Admin angemeldet. Es existiert mindestens ein User-Account auf dem System.
- **Ablauf:**  
Der Admin klickt auf den Reiter „Manage User“. Dann wählt er den zu löschenden User aus und drückt auf den dafür gekennzeichneten Button.
- **Erfolg:**  
Der User wird aus der Datenbank gelöscht und wird nicht mehr auf der Seite „Manage Users“ angezeigt. Der gelöschte User kann sich natürlich nicht mehr mit seinem Account anmelden.
- **Kein Erfolg:** Der Admin erhält eine Fehlermeldung und der User bleibt weiterhin in der Datenbank gespeichert.
- **Involvierte Klassen:** Userx

## Status der AccessPoints/SensorStations einsehen:

- **Vorbedingung:**  
Das System funktioniert fehlerfrei, es gibt mindestens eine SensorStation bzw. AccessPoint, die bereits verbunden sind. Der aktuelle Nutzer ist als Admin angemeldet.
- **Ablauf:**  
Der Admin klickt auf den Reiter „Devices“ und bekommt dann eine Übersicht über den Status der Geräte.
- **Erfolg:**  
Der Admin bekommt alle Geräte zu sehen und kann sich einen Überblick über den Status verschaffen.
- **Kein Erfolg:**  
Die Seite „Devices“ kann nicht aufgerufen werden, der Admin bekommt eine Fehlermeldung und muss es nach einer Zeit wieder probieren.
- **Involvierte Klassen:**  
Userx, AccessPoint, SensorStations

## Logs einsehen:

- **Vorbedingung:**  
Das System funktioniert und der Akteur ist als Admin angemeldet.
- **Ablauf:**  
Der Admin klickt auf den Reiter „Logs“.
- **Erfolg:**  
Der Admin hat Einsicht über alle AuditLog-Einträge.
- **Kein Erfolg:**  
Die Seite „Logs“ ist nicht verfügbar. Der Admin bekommt eine Fehlermeldung.
- **Involvierte Klassen:**  
Userx, AuditLog

## 2.3.4 Akteur: Visitor

### Bilder hochladen:

- **Vorbedingung:** Der Besucher befindet sich beim Gewächshaus und scannt den QR-Code
- **Ablauf:** Nach dem Scan befindet sich der Besucher auf der Seite, auf der sich eine Bildgalerie der gescannten Pflanze befindet. Der Besucher kann nun ein Bild hochladen in dem er auf den dazugehörigen Button klickt.
- **Erfolg:** Das neue Bild befindet sich in der Bildgalerie und jeder, der mit dieser Bildgalerie interagiert, kann das neue Bild betrachten.
- **Kein Erfolg:** -
- **Involvierte Klassen:** Userx, PlantPicture, SensorStation

### Bilder anschauen:

- **Vorbedingung:** Der Besucher befindet sich beim Gewächshaus und scannt den QR-Code
- **Ablauf:** Nach dem Scan befindet sich der Besucher auf der Seite, auf der sich eine Bildgalerie der gescannten Pflanze befindet.
- **Erfolg:** Der Besucher kann sich nun durch die Bildgalerie forsten und die Bilder anschauen.
- **Kein Erfolg:** Eine Fehlermeldung wird angezeigt
- **Involvierte Klassen:** Userx, PlantPicture, SensorStation



### 2.3.5 Akteur: AccessPoint

#### Verbindung zwischen SensorStation und AccessPoint herstellen:

- **Vorbedingung:** Webserver läuft, Admin eingeloggt, AccessPoint mit Webserver verbunden
- **Ablauf:**
  - Admin setzt AccessPoint via Webinterface in Kopplungsmodus.
  - SensorStation mithilfe von Taster auf Mikrocontroller in Kopplungsmodus setzen. LED fängt an, blau zu blinken.
  - SensorStation findet einen (oder mehrere) AccessPoints im Kopplungsmodus.
  - SensorStation verbindet sich mit dem ersten AccessPoint. Kopplungsmodus von SensorStation wird beendet. LED hört auf zu blinken.
  - Admin gibt SensorStation manuell im Webinterface frei.
  - Admin beendet den Kopplungsmodus des AccessPoints via Webinterface.
- **Erfolg:** SensorStation mit AccessPoint verbunden, keines der beiden Geräte im Kopplungsmodus, Kommunikation möglich
- **Kein Erfolg:**
  - Kein AccessPoint in der Nähe. Nach fünf Minuten wird Kopplungsmodus automatisch beendet.
  - Falls die ID der SensorStation bereits von einer anderen mit dem AccessPoint verbundenen Station verwendet wird, verbindet sich die SensorStation mit einem anderen AccessPoint. Falls keiner vorhanden wird der Kopplungsmodus beendet.
- **Involvierte Klassen:** Admin, AccessPoint, SensorStation

#### Accesspoint sendet Messdaten an Webserver:

- **Vorbedingung:** Accesspoint und die Sensorstation ist auf der Webseite registriert. Accesspoint hat die Daten erfolgreich erhalten und zwischengespeichert.
- **Ablauf:** Accesspoint sendet die Daten über eine REST-schnittstelle an den Webserver und wartet auf die Rückmeldung ob die Daten empfangen wurden. Der Webserver empfängt die Daten und sendet dem Accesspoint eine Bestätigung. Nach Erhalt der Bestätigung löscht der Accesspoint die Messdaten.
- **Erfolg:** Der Webserver erhält die Daten und der Accesspoint erhält die Bestätigung.
- **Kein Erfolg:** Der Accesspoint behält die Daten im Speicher und sendet die Daten erneut.
- **Involvierte Klassen:** AccessPoint, SensorStation

### Wertüberschreitung mitteilen:

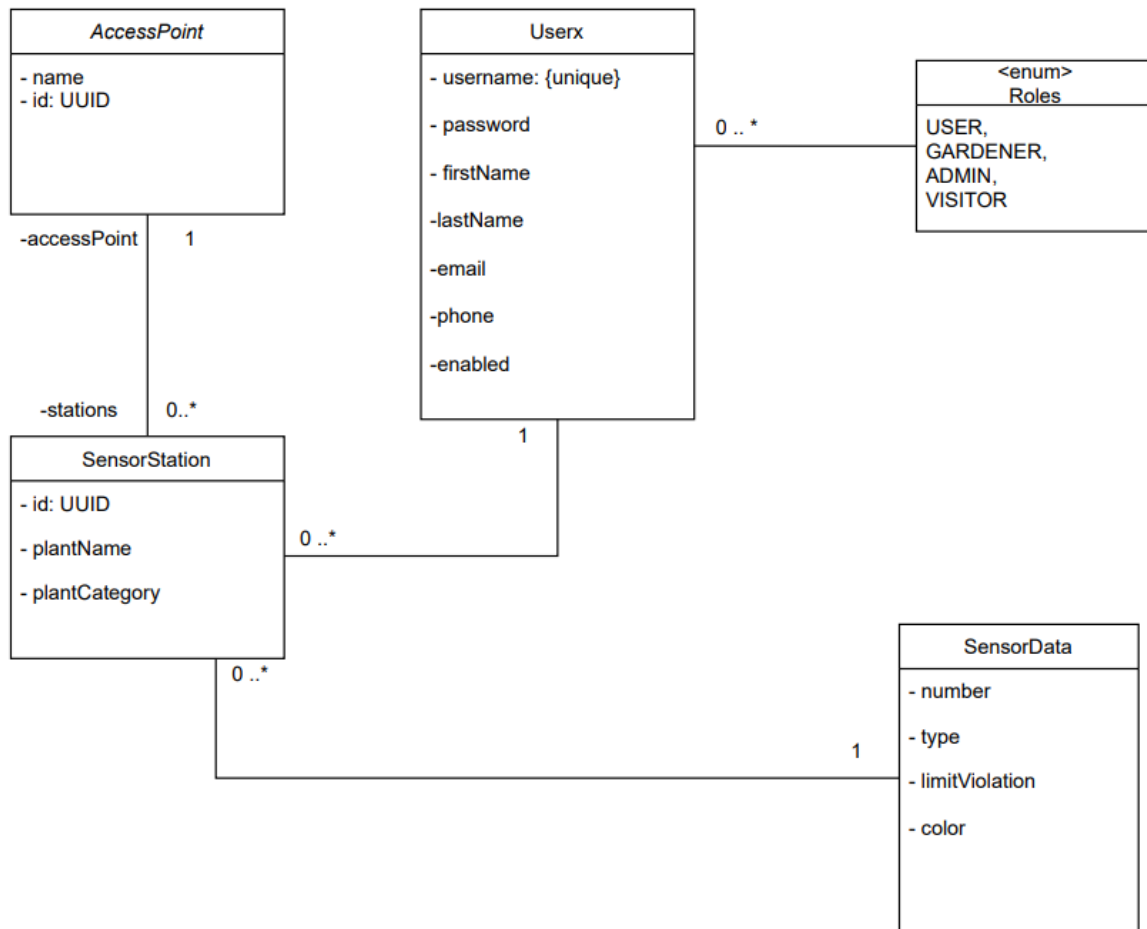
- **Vorbedingung:** Webserver läuft, Accesspoint und SensorStation mit Server verbunden, Wertüberschreitung für konfiguriertes Intervall festgestellt
- **Ablauf:**
  - AccessPoint sendet die Intensität der Überschreitung und den betroffenen Wert an die SensorStation, bei der die Überschreitung festgestellt worden ist.
  - Die SensorStation gibt anhand des Typs und anhand der Intensität einen Farbton in einem gewissen Blinkcode aus.
  - AccessPoint sendet die Intensität, den betroffenen Wert und die betroffene SensorStation an den Webserver, welcher die Grenzwertüberschreitung in einer Nachricht anzeigt.
- **Erfolg:** LED der SensorStation blinkt und am Webserver wird eine Nachricht angezeigt
- **Kein Erfolg:** Die SensorStation erhält die Message des AccessPoint nicht und kann die Über-/Unterschreitung nicht ausgeben.
- **Involvierte Klassen:** Accesspoint, SensorStation

### 2.3.6 Akteur: SensorStation

#### Werte an AccessPoint senden:

- **Vorbereitung:** Verbindung zwischen AccessPoint und Sensorstation sind aktiv und haben eine aktive Verbindung.
- **Ablauf:** Die Sensorstation sendet Daten über Bluetooth LE an den AccessPoint
- **Erfolg:** Der AccessPoint erhält die Daten, kann sie verarbeiten und ggf. Werteüber-/unterschreitungen an die Sensorstation senden.
- **Kein Erfolg:** Die Verbindung zwischen AccessPoint und Sensorstation bricht ab und die Daten können nicht geschickt werden. Der AccessPoint meldet die Sensorstation als offline an den Webserver.
- **Involvierte Klassen:** Accesspoint, SensorStation

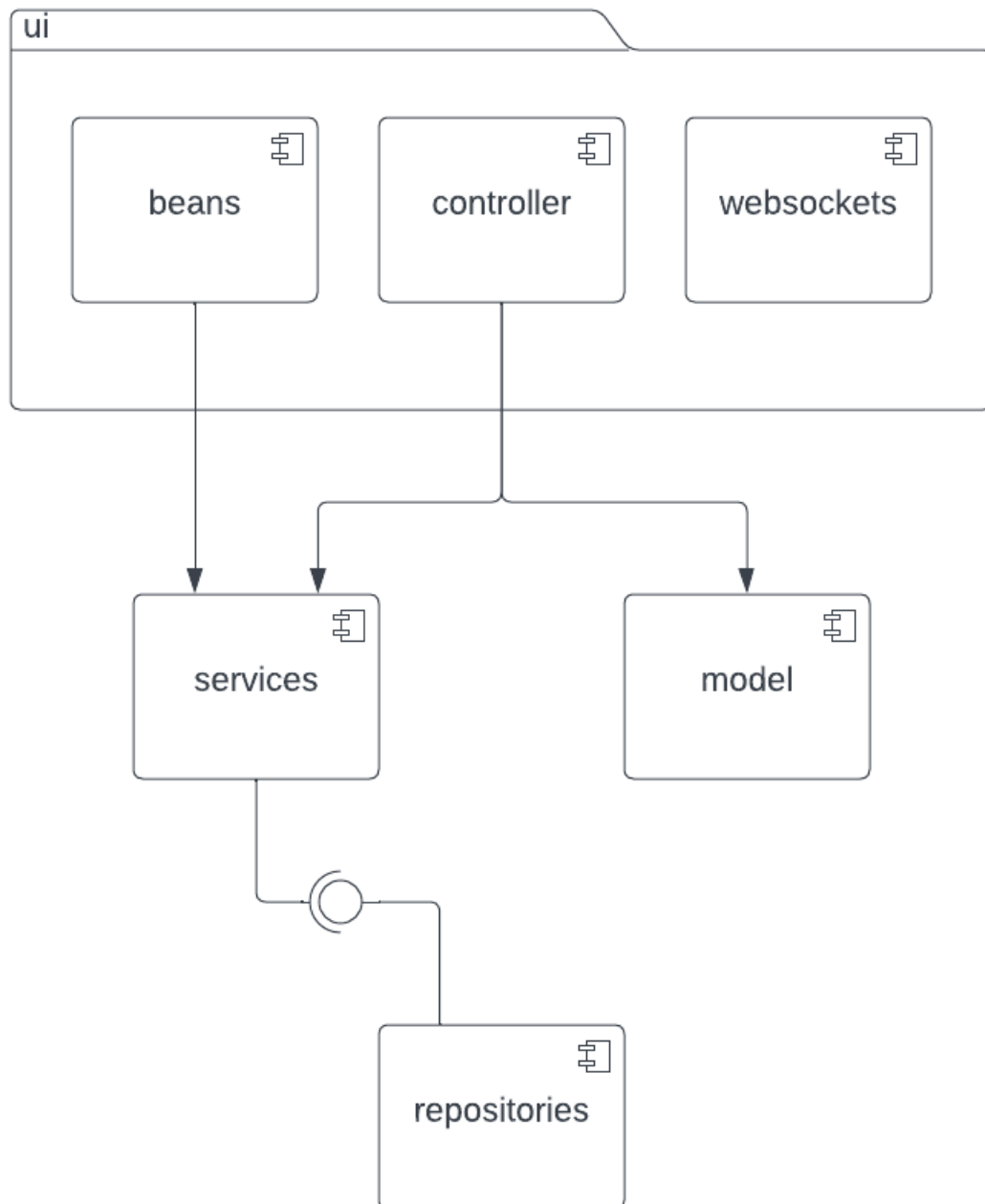
### 3 Fachliches Klassendiagramm



Userx ist die Basisklasse für alle vier möglichen Nutzer, diese werden über ein Set identifiziert, in dem das Enum "Roles" steht. Einer SensorStation ist genau einem AccessPoint zugeordnet, jeder AccessPoint kann so viele Sensorstationen betreuen, wie es seine Verbindung zulässt. Die Klasse SensorData enthält den Typen der Daten (Wassergehalt, Luftqualität, usw.), den Messwert selbst, einen Wert, ob der Grenzwert überschritten wurde und die Farbe, in der die LED blinken soll, als String.

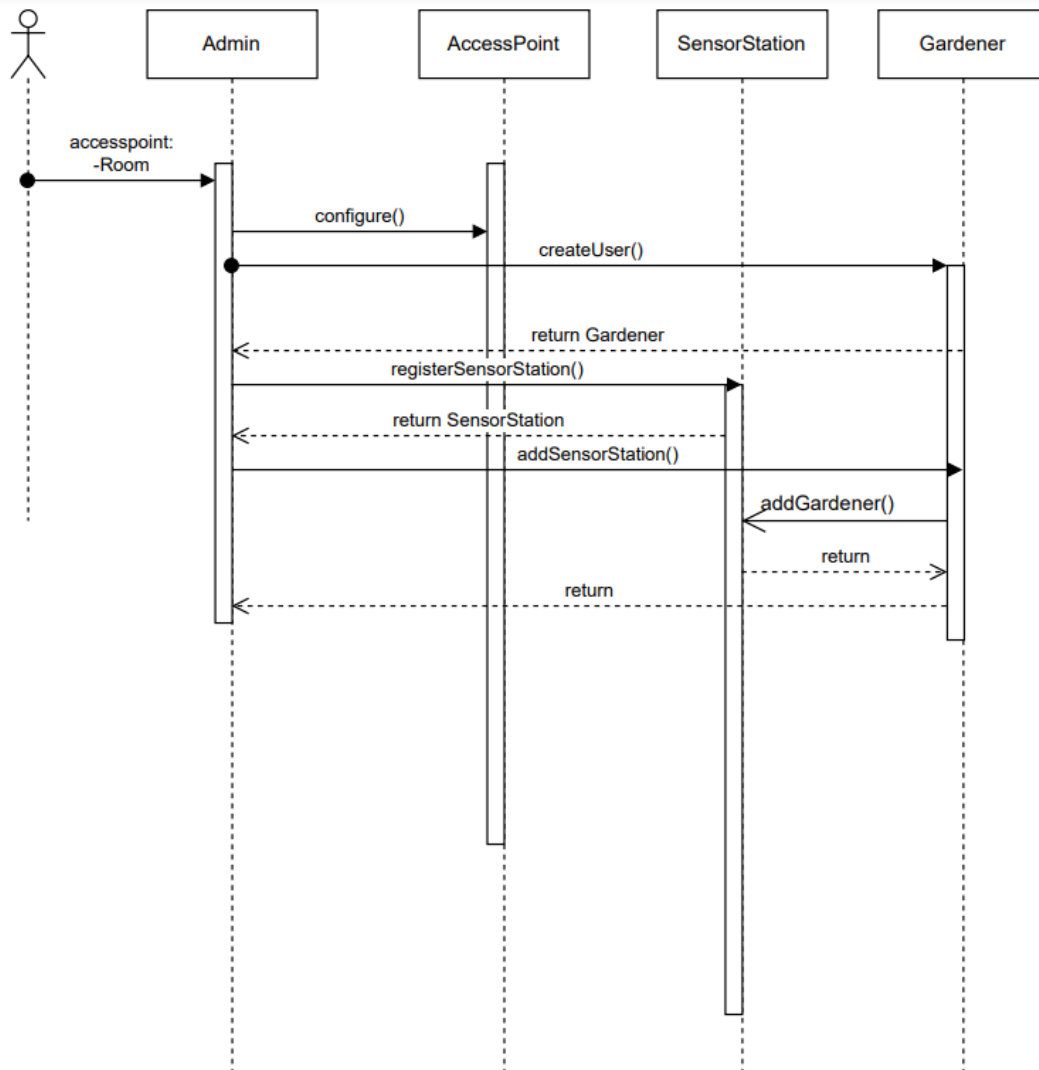
## 4 Software-Architektur

### 4.1 Komponentendiagramm

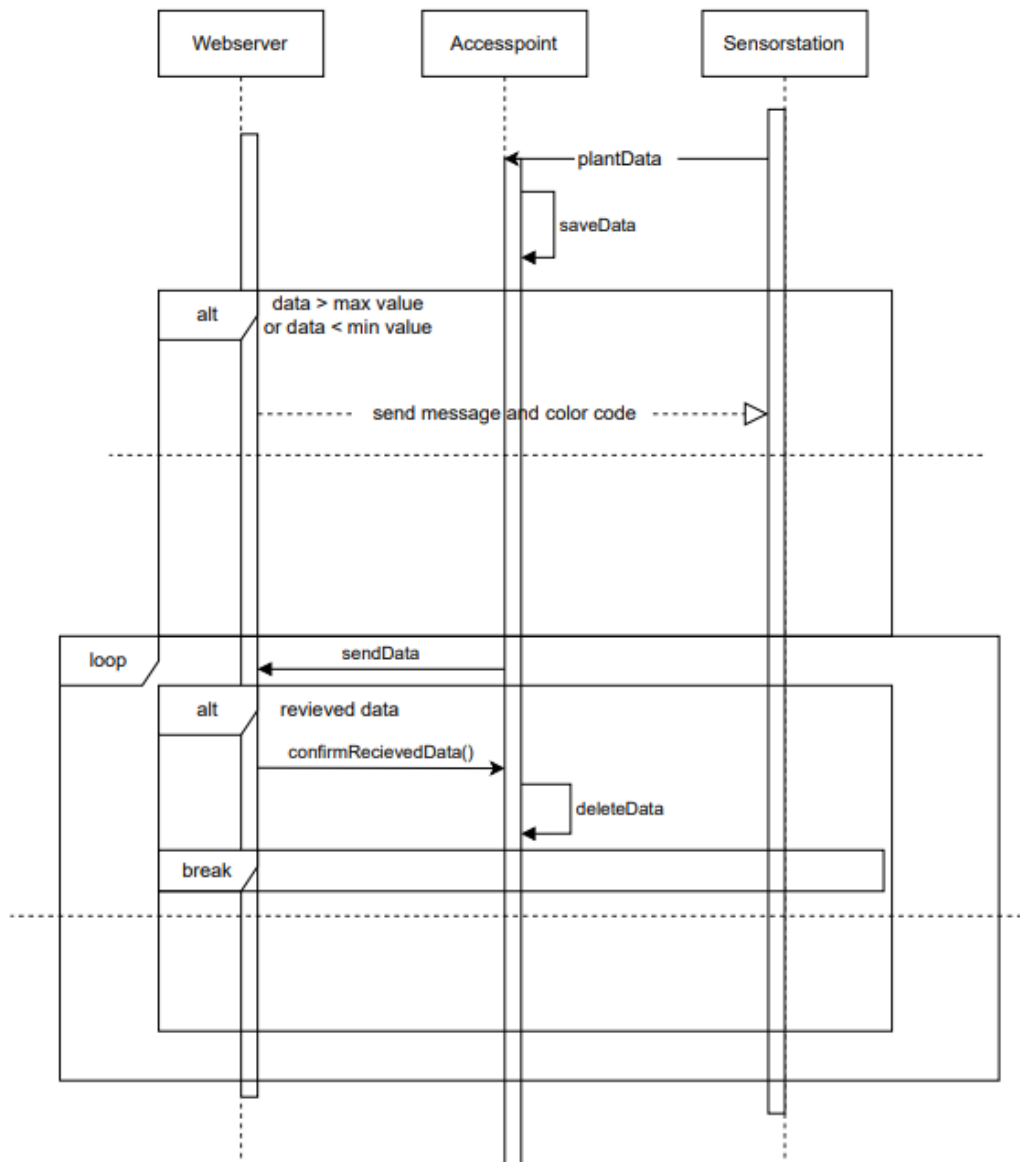


## 4.2 Laufzeitsicht

### Ablauf Einrichtung der Systeme:



## Ablauf Datenübertragung zwischen den Systemen:



## 4.3 Ausgewählte Technologien

### **Java:**

Der Webserver wird mit der objektrelationalen Programmiersprache Java implementiert.

### **C:**

C ist eine Programmiersprache, die für die Entwicklung von Betriebssystemen, Gerätetreibern und Systemsoftware entwickelt wurde. Die Anwendung auf dem Arduino wird mit C implementiert.

### **Python:**

Python ist eine Programmiersprache, die vor allem durch ihre große Auswahl an Bibliotheken besticht. Die Anwendung auf unserem Raspberry Pie (dem AccessPoint) wird mit Python implementiert, da sich damit die Kommunikation leichter implementieren lässt.

### **Spring:**

Spring ist ein Framework für Java. Die Technologie bietet eine umfassende Infrastruktur zur Entwicklung von Anwendungen auf Basis von Inversion of Control und Dependency Injection. Zusätzlich wird Spring Boot verwendet.

### **MySQL:**

MySQL ist ein relationales Datenbanksystem. Mit MySQL lassen sich Daten speichern, abfragen und löschen. Zusammen mit Java ist ORM möglich.

### **GitLab:**

Gitlab basiert auf dem Versionskontrollsystem Git, dies ermöglicht eine effiziente Lösung, um als Team an dem Projekt zu arbeiten.

### **Sonar-Qube:**

Sonar-Qube ist ein Analyse-Tool und wird mit dem Repository auf Git verknüpft. Sonar-Qube erleichtert die Analyse von Code Smells, Bugs und Testabdeckung.

### **JUnit:**

JUnit ist ein Test-Framework für Java. Mit diesem Framework werden automatisierte Tests für Java-Anwendungen implementiert.

**Java Server Faces:**

JSF ist ein Framework und erleichtert die Entwicklung von Weboberflächen. JSF wird für die Webdarstellung des Webservers verwendet.

**Primefaces:**

Primefaces ist ein Framework und erweitert eine JSF-Implementierung.

**Apache Maven:**

Apache Maven ist für das Build-Management zuständig. Mit dieser Technologie werden Java-Programme verwaltet.

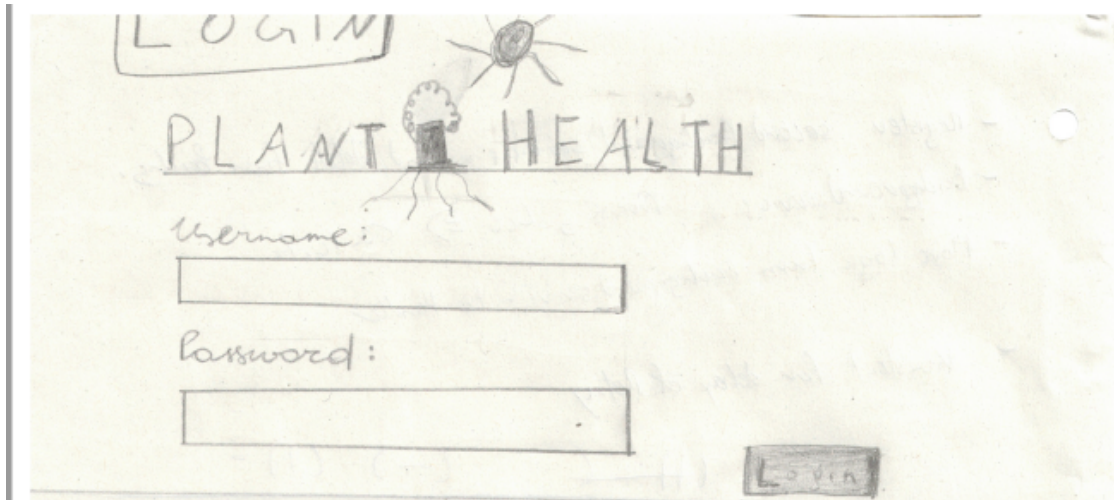


## 5 Ausfallsicherheit

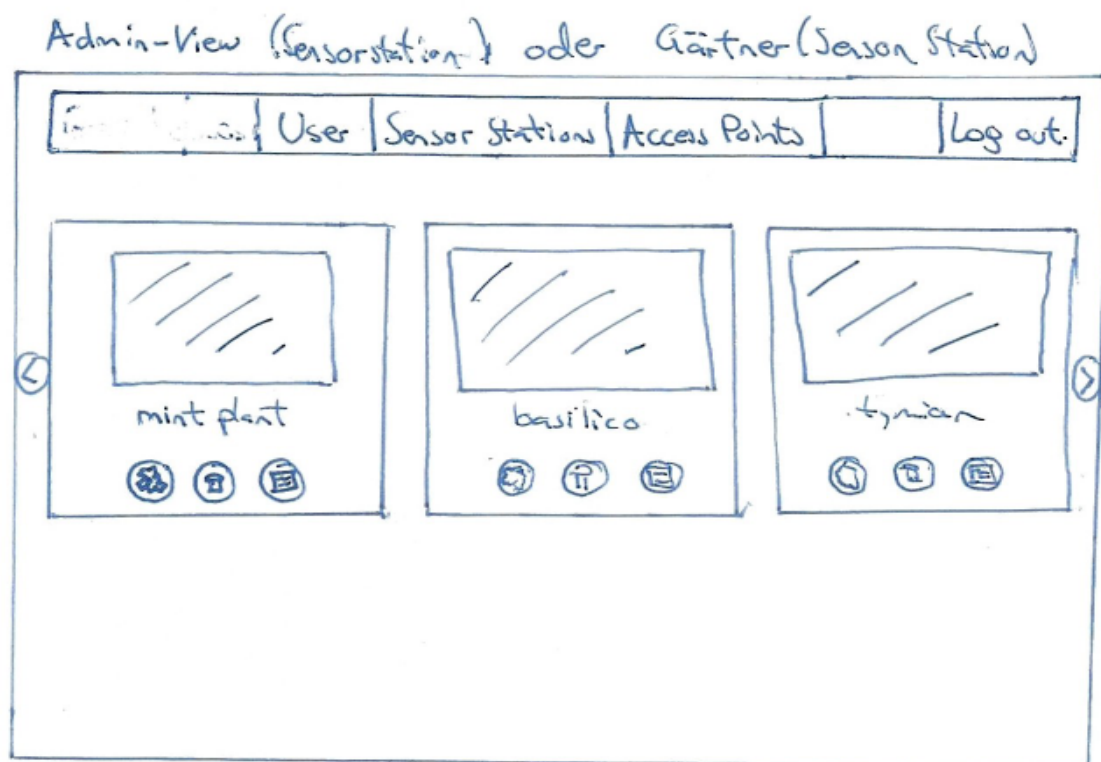
- T1: Eingeschränkte Kommunikation zwischen Arduino Sensorstationen und Minirechner
  - Daten, die von dem Arduino kommen, werden nicht in die Datenbank aufgenommen, da sie u.U. verfälscht sind
  - Warten, bis Verbindung besser wird
- T2: Unerwarteter Neustart eines Minirechners (Raspberry)
  - muss gewährleistet sein, dass alle Sensorstationen, die vor dem Neustart verbunden waren, wieder eine Verbindung aufgebaut wird
  - Skripte zur automatischen Wiederherstellung der Verbindung
  - Daten, die in dieser Zeit zum Raspberry gesendet wurden, sind verloren
- T3: Temporärer Ausfall der Kommunikationswege zwischen Minirechnern und zentralem Backend
  - Daten (bis zu einem gewissen Punkt) puffern bis Verbindung wiederhergestellt wurde
  - automatische Wiederherstellung der Verbindung
- T4: Kurzfristiger Ausfall des zentralen Backends.
  - Benachrichtigung alle Benutzer über den den Ausfall
- T5: Ausfall der Sensorstation
  - Verbindung zum AccessPoint ist eingeschränkt, aber Stromversorgung ist vorhanden:
    - optisches und akustisches Signal (blinken)
  - kompletter Ausfall des Strom:
    - AccessPoint sendet Nachricht an den Webserver, ein Admin sieht diese Fehlermeldung und muss dann die Sensorstation warten und gegebenenfalls neu einrichten.

## 6 GUI Prototyp

Login-Page für alle Rollen:



Sicht des Admins auf die Sensorstationen bzw. Sicht des Gärtners auf die ihm zugeteilten Stationen:



Sicht des Admins auf die User-Daten und mögliche Verwaltung:

### Admin-View (Users)

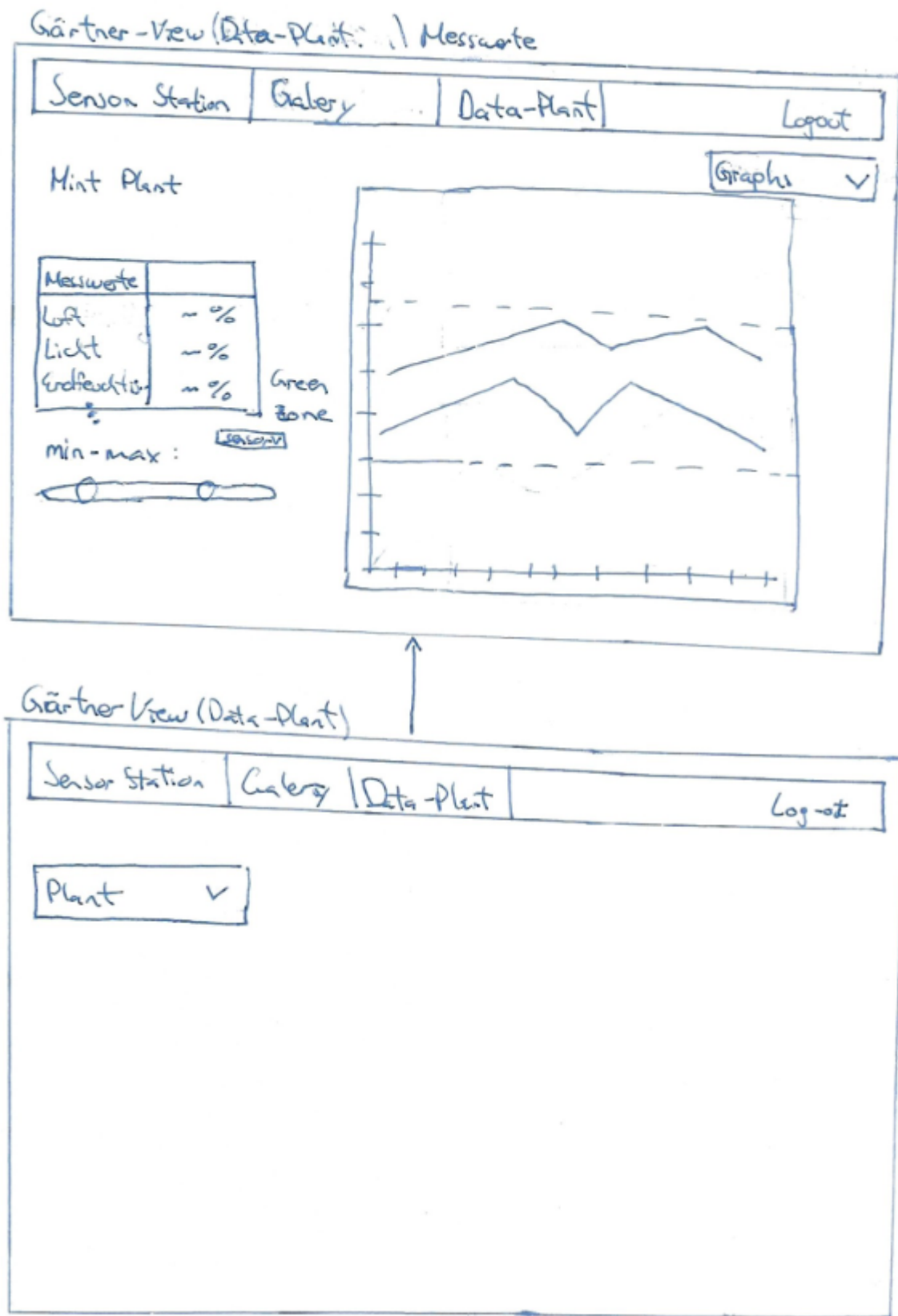
Admin-View (Users)				
Navigation	User	Sensor Station	Access Point	Logout
Username	First Name	Last Name	enabled	
Max M	Max	Mustermann	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Optimal
				<input type="checkbox"/>
				<input type="checkbox"/>
				<input type="checkbox"/>

Sicht des Admins auf die Access Points und deren Verwaltung:

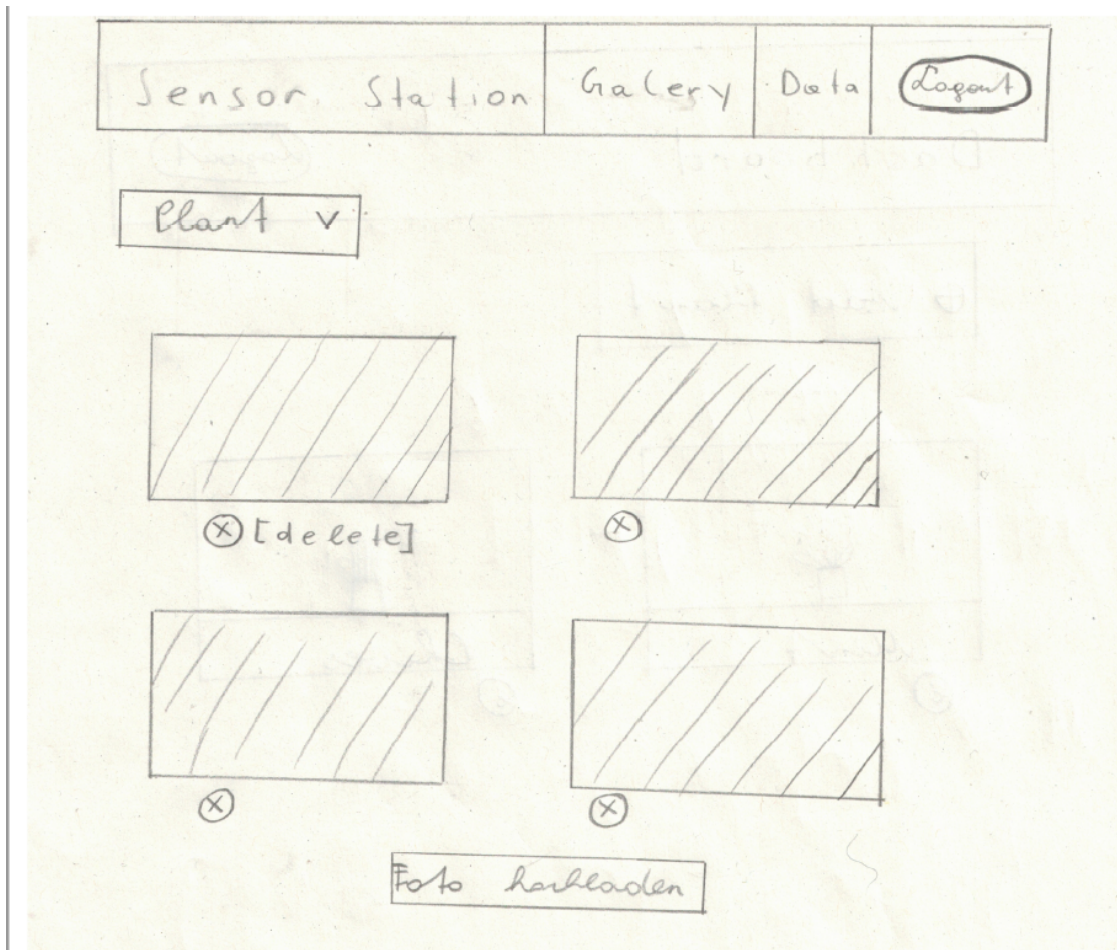
### Admin-View (Access Points)

Admin-View (Access Points)			
User	Sensor Station	Access Points	Logout
Access Points-key		State	
			<input checked="" type="radio"/>
			<input type="radio"/>
			<input type="radio"/>
			<input type="radio"/>
			<input type="radio"/>

Sicht des Gärtners auf die Messwerte und der Daten der Pflanzen:

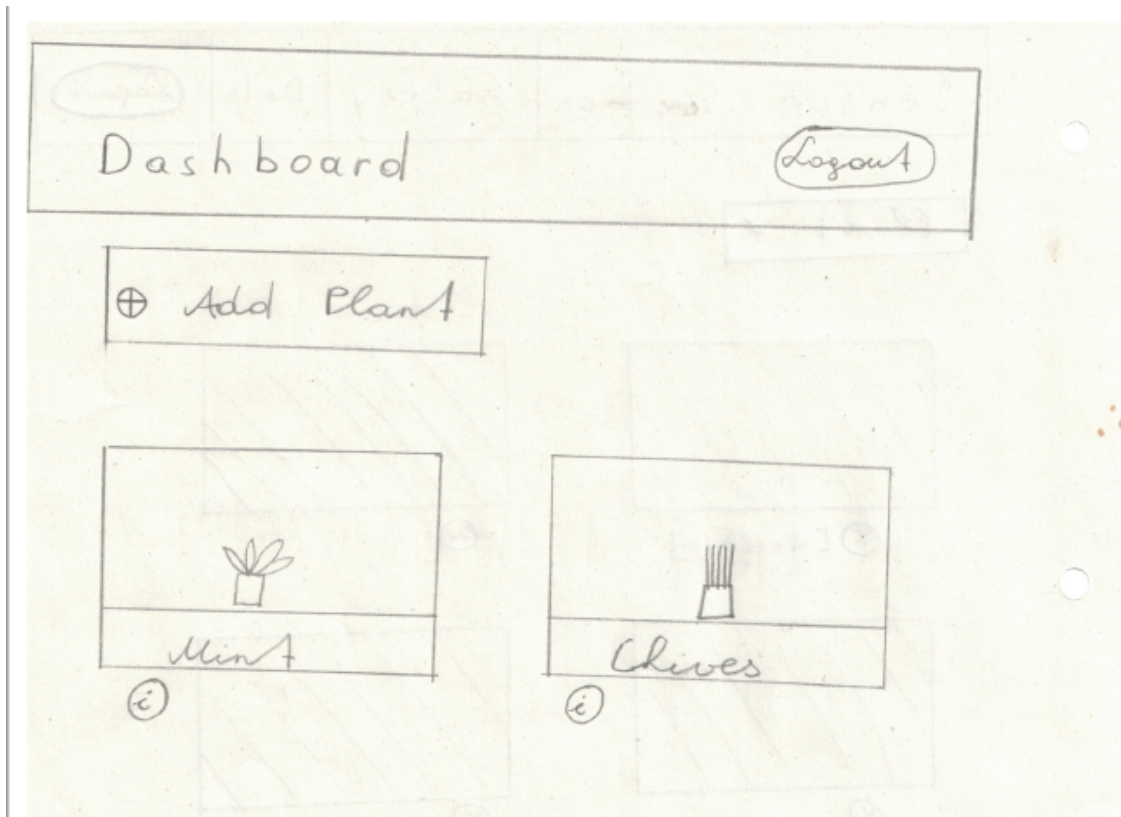


Sicht des Gärtners auf die Bildergalerien seiner Pflanzen:

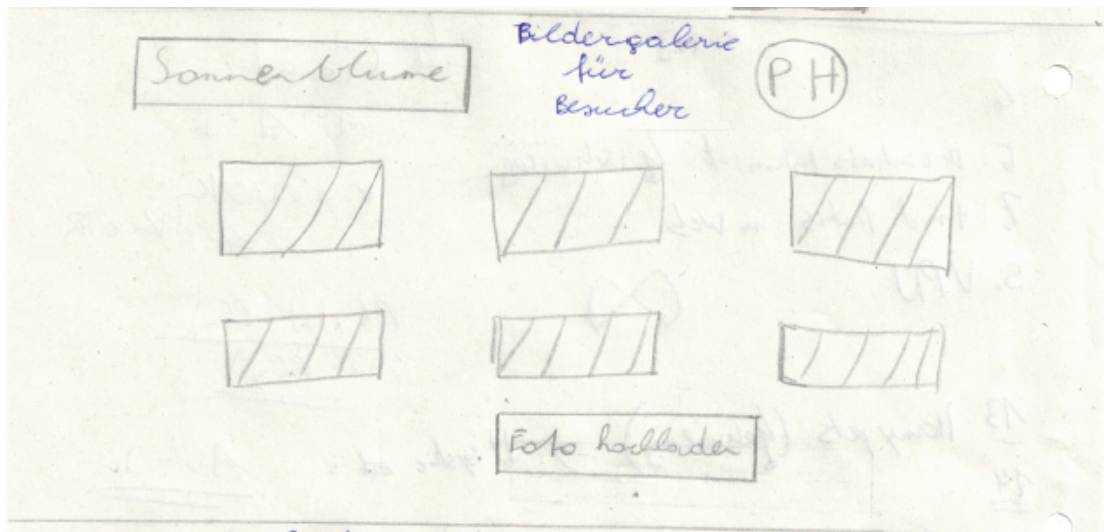




Sicht des Nutzers auf sein Dashboard:



Sicht des Besuchers (über den QR-Code):



# 7 Projektplan

## 7.1 Verantwortlichkeiten

Marco Cotrotzo:

- REST API mit Spring Framework,
- Testdrehbücher und Abnahmetests,
- Java Webapp

Benedikt Glaser:

- Arduino
- Python, REST und Co @ Raspberry Pi
- Kontaktperson

Medin Jasharoski:

- REST API mit Spring Framework,
- Testdrehbücher und Abnahmetests,
- Java Webapp

Andreas Kirchmair:

- Github
- Arduino
- Bluetooth

Jona Müller:

- Python, REST und Co @ Raspberry Pi
- Bluetooth
- Java Webapp

## 7.2 Milestones

- 16. 3 Einrichtung Git
- 16.3 Konzept fertig
- 23.3 Implementierung fachliche Komponenten (Models) zwischen Raspberry & Webserver
- 23.3 Steckplan fertig
- 30.3 REST-Schnittstellen
- 6.4 Bluetooth Verbindung zw. Arduino & Raspberry
- 20.4 Hardwareaufbau fertig
- 10.5 Lauffähiges Projekt
- 18.5 Bugfixes & Implementation
- 19.5 Lauffähiges Projekt + Präsentation
- 24. 5 Bugfixes & Implementation
- 25.5 Abnahmetests
- 20.6 Bugfixes & Dokumentierung (ARC 42)
- 22.6 finale Abgabe + Präsentation