Analysis Document

By AutoMate



Names: Sean Brokke - 4208501

Mitchell van ‘t Kruys - 4058151

Daniel García López -4976215

Nick Mulders - 4523504

Andre Sanao -4425553

Luka Aerts - 4202317

Mentors: S. Andova

Project: Industry Project semester 3

Due date: 31st of October 2022

**Table of Contents**

[Analysis 3](#_Toc119924082)

[Problem to solve 3](#_Toc119924083)

[Use Case 3](#_Toc119924084)

[User requirements 4](#_Toc119924085)

[Functional requirements 5](#_Toc119924086)

[MoSCoW Analysis 7](#_Toc119924087)

[Design 8](#_Toc119924088)

[Description 8](#_Toc119924089)

[Component diagram 8](#_Toc119924090)

[Communication between components 8](#_Toc119924091)

[Context diagrams 10](#_Toc119924092)

[Plant Sensor Module 10](#_Toc119924093)

[Water Tank Sensor Module 10](#_Toc119924094)

[Actuators Module 11](#_Toc119924095)

[Gateway Module 11](#_Toc119924096)

[UML diagram 11](#_Toc119924097)

[Hardware description 11](#_Toc119924098)

[Description application protocol 15](#_Toc119924099)

[Communication Protocol 15](#_Toc119924100)

[State diagrams 16](#_Toc119924101)

[Node-Red: 16](#_Toc119924102)

[Plant sensor Module: 17](#_Toc119924103)

[Actuator Module: 17](#_Toc119924104)

[Water valve Module: 17](#_Toc119924105)

[Gateway Module: 17](#_Toc119924106)

[Proof of Concepts 18](#_Toc119924107)

[Performance considerations 18](#_Toc119924108)

[Flow charts 19](#_Toc119924109)

[Test Plan 20](#_Toc119924110)

[Description 20](#_Toc119924111)

[Specifications 21](#_Toc119924112)

[Components 21](#_Toc119924113)

[Sensors 21](#_Toc119924114)

[Temperature & Humidity sensor 21](#_Toc119924115)

[Moisture sensor 21](#_Toc119924116)

[UV sensor 21](#_Toc119924117)

[Water Level sensor 22](#_Toc119924118)

[Actuators 22](#_Toc119924119)

[Servo (valve of water) 22](#_Toc119924120)

[Fan (For better temperature levels) 22](#_Toc119924121)

[LED /UV light (For better temperature levels) 22](#_Toc119924122)

[Humidifier (For better humidity levels) 22](#_Toc119924123)

[References 22](#_Toc119924124)

# Analysis

## Problem to solve

Our vision is to create solutions that make daily tasks simple. One can think of the systems approach as an organized way of dealing with a problem. The system will have a collection of components that work together to realize our objective. The objective is to simulate an automated greenhouse system, where smart embedded systems would maintain multiple plants health.

The problem we as a company are trying to solve is that for a farmer, it is very hard to keep track of his plants if the amount is scaling up to tens of thousands.  
Our system could also be helpful for ordinary people who do not take an interest in taking care of their plants but do like looking at them.

## Use Case

|  |  |
| --- | --- |
| **Use Case ID: 001** | **Use Case:** Monitor the plants health |
| **Description** | The system should be able to measure the plants health by reading all values from multiple sensors. |
| **Actor**: | The system |

|  |  |
| --- | --- |
| **Use Case ID: 002** | **Use Case:** Display data on a dashboard |
| **Description** | The system should display applicable data, like measurements or device status on a neat and clear dashboard. |
| **Actor**: | The system |

|  |  |
| --- | --- |
| **Use Case ID: 003** | **Use Case:** Notify the user |
| **Description** | The system should notify the user in case he/she has to perform an action if the system is not able to. |
| **Actor**: | The system |

|  |  |
| --- | --- |
| **Use Case ID: 004** | **Use Case:** Activate actuators |
| **Description** | The system should active its actuators to keep the plant in optimal conditions. |
| **Actor**: | The system |

## User requirements

|  |  |  |
| --- | --- | --- |
| **User Requirement ID** | **Description** | **Use Case** |
| **UR**\_001 | The user shall be able to examine all measured values on a clear dashboard. | **UC**\_001 / **UC**\_002 |
| **UR**\_002 | The user shall be able to access the dashboard at any location at any time. | **UC**\_002 |
| **UR**\_003 | The user shall be able to be notified after a device has broken down or after the communication has failed. | **UC**\_003 |
| **UR**\_004 | The user shall be able to understand what he/she needs to do after a notification has appeared. | **UC**\_003 |
| **UR**\_005 | The user should be notified if he/she must refill the water tank if the system has notified that the tank is empty. | **UC**\_001 / **UC**\_003 |

## Functional requirements

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Description** | **Use Cases** |
| **FR**\_001 | The system must be able to measure temperature, humidity in a room. | **UC**\_001 |
| **FR**\_002 | The system must be able to measure moisture and UV levels per plant. | **UC**\_001 |
| **FR**\_003 | The system must be able to display the measured values on a dashboard. | **UC**\_001 / **UC**\_002 / **UR**\_001 |
| **FR**\_004 | The system must notify the user in case the user has to perform an act that the system is not able to do on its own. | **UC**\_003 / **UC**\_002 / **UR**\_003/ **UR**\_004 / **UR**\_005 |
| **FR**\_005 | The system must notify the user in case a device breaks. | **UC**\_003 / **UC**\_002 / **UR**\_003/ **UR**\_004 |
| **FR**\_006 | The system must regulate the temperature in the room between desired thresholds by turning on a fan. | **UC**\_001 |
| **FR**\_007 | The system must be able to maintain the soil’s moisture between desired thresholds. | **UC**\_004 / **UC**\_001 |
| **FR**\_008 | The system must be able to maintain the temperature in between a max and minimum threshold. | **UC**\_004 / **UC**\_001 |
| **FR**\_009 | The system should be able to notify the user when the water tank is empty, or the minimum threshold is reached. | **UC**\_003 / **UC**\_005 / **UR**\_003 / **UR**\_005 |
| **FR**\_010 | The system must be able to communicate with each other at any point in time | **UC**\_001 |
| **FR**\_011 | The system must be able to maintain the light level between the desired thresholds. | **UC**\_004 / **UC**\_001 |
| **FR**\_012 | The system must pick up connection from another device in case that system has broken down. | **UC**\_001 |
| **FR**\_013 | The system must have a comprehensible UI. | **UC**\_002 |
| **FR**\_014 | The system must make use of the humidifier in case the air is dry | **UC**\_004/ **UC**\_001 |
| **FR**\_015 | The system must make use of the fan in case the air quality is damaging the plant | **UC**\_004/ **UC**\_001 |
| **FR**\_016 | The system must have a stable communication between devices so it can share its measurements | **UC**\_001 |
| **FR**\_017 | The system should react on the environment of the plant by activating its actuators | **UC**\_001 |
| **FR**\_018 | The water tank for the plants is below the threshold. The system should notify the user and the user should refill. | **UC**\_001 / **UC**\_003  UR\_005/ UR\_004 |
| **FR**\_019 | The system must always know which devices still have connection | **UC**\_001 |

## MoSCoW Analysis

A picture containing treemap chart

Description automatically generated

**Should Have**

1. Notifications

In case a plants conditions are bad; the user will be notified by the dashboard. Think of messages like:  
“Your plant is lacking sunlight”, or   
“Your plant is dry, give it some water”.

1. Error Handling

The error handling is important in case the system breaks. If this happens it prevents the system from crashing.

1. Easy initialization method

Our system should be accessible for every user, even people who are digital literate. Therefore, we should make our user interface as comprehensible as possible.

1. Communication

We will implement a wireless communication via MQTT. This is important because our system will be an integrated system.

1. Sensors

The sensors are important to monitor the plants conditions in real-time. For a list of our sensors, we refer to our components list.

1. Dashboard  
   Our monitored values for the sensors will be displayed on the dashboard. This is important for our users, so they have a nice and clear view of their plant’s conditions.
2. Actuators  
   The actuators will react if certain sensor values deviate from a set threshold. After the first iteration is finished, we will add these   
   actuators to the system. For a list of our sensors, we refer to our components list.

Shape

Description automatically generated with low confidence**Shape

Description automatically generated with low confidence**

**ard**

# Design

## Description

1. Short Description of the proposed solution for the problem**(daniel)**

Our vision is to create solutions to make daily tasks simple. One can think of the systems approach as an organized way of dealing with a problem. The system will have a collection of components that work together to realize our objective. The objective is to simulate an automated greenhouse system, where smart embedded systems would maintain multiple plants health.

The greenhouse is structured like that we have a dashboard which is in constant communication with the esp32s,responsibles of the different components of the greenhouse(UV, sensors, moisture, sensors etc.).We have to have communication also between the different esp32.

## Component diagram

### Communication between components

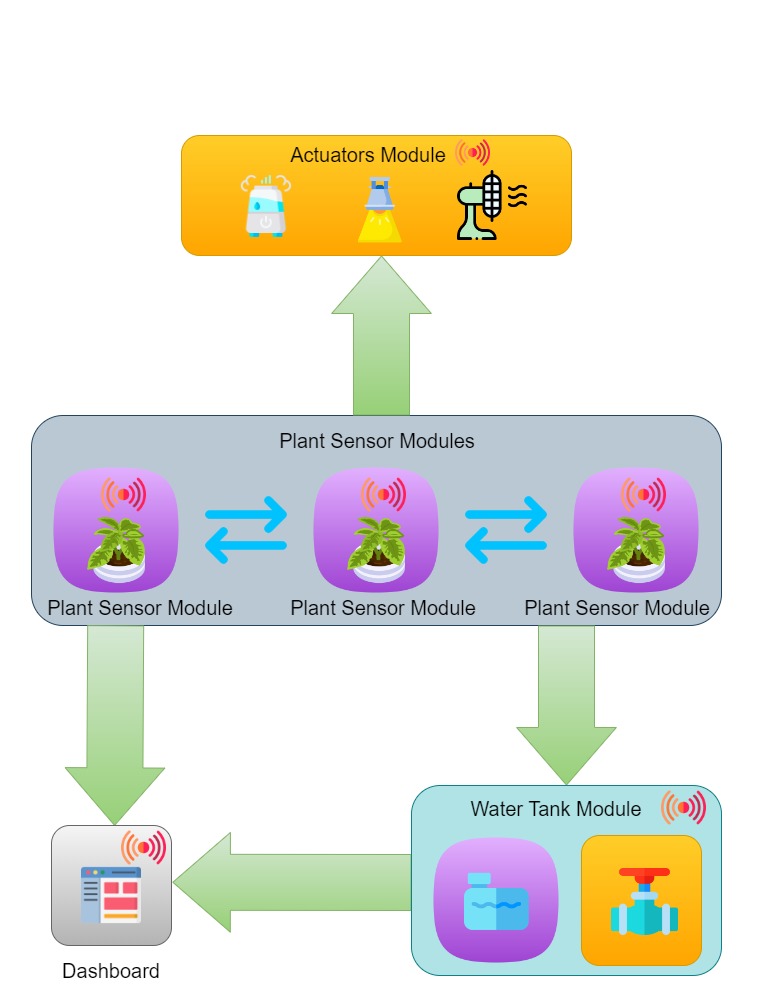


Figure 1 Communication between components

|  |  |  |
| --- | --- | --- |
| Module: | Components: | Communicate: |
| Plant Sensor Module | **Sensors:**   * Soil Moisture Sensor * UV Light Sensor * Temperature & Humidity Sensor   **Actuators:** | The Plant Sensor Module can share its sensors values with other plant sensor modules. This module also sends sensor values to the dashboard. This module also sends commands to the Actuators Module to set appropriate UV Light, Fan speed and Humidifier settings. This module also sends commands to the Water Tank Module to open or close water valve when in need of moisture in the soil. |
| Water Tank Module | **Sensors:**   * Water Level Sensor   **Actuators:**   * 180-degrees Servo (Valve) | The Water Sensor Module sends its water level to the dashboard and receives commands from the Plant Sensor Modules to either open or close its water valve. |
| Actuators Module | **Sensors:**  **Actuators:**   * UV Light (Yellow LED) * Fan (5v Fan) * Humidifier (Blue LED) | The Actuator Module receives commands from the Plant Sensor Modules, to set appropriate UV Light, Fan speed and Humidifier settings. |

## Context diagrams

### Plant Sensor Module

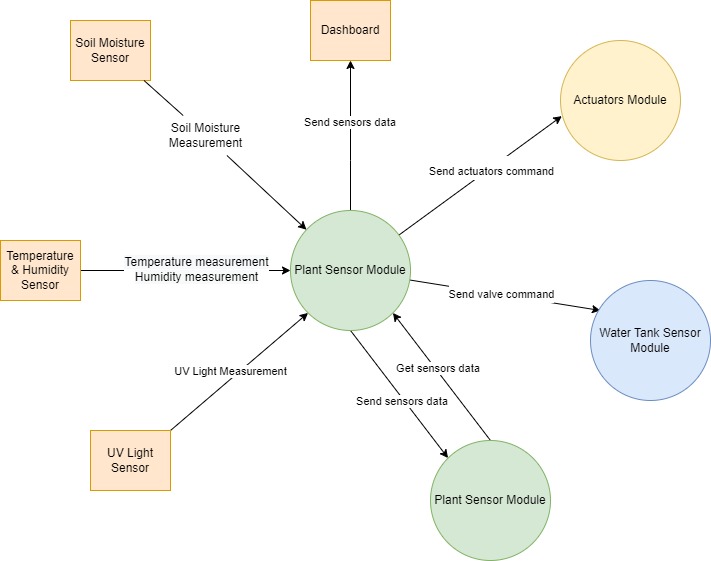


Figure 2 Plant sensor module context diagram

### Water Tank Sensor Module

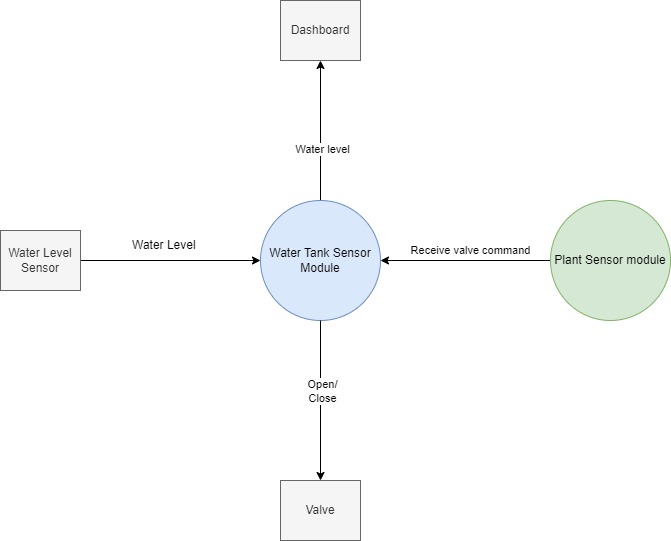


Figure 3 Water tank sensor module context diagram

### Actuators Module

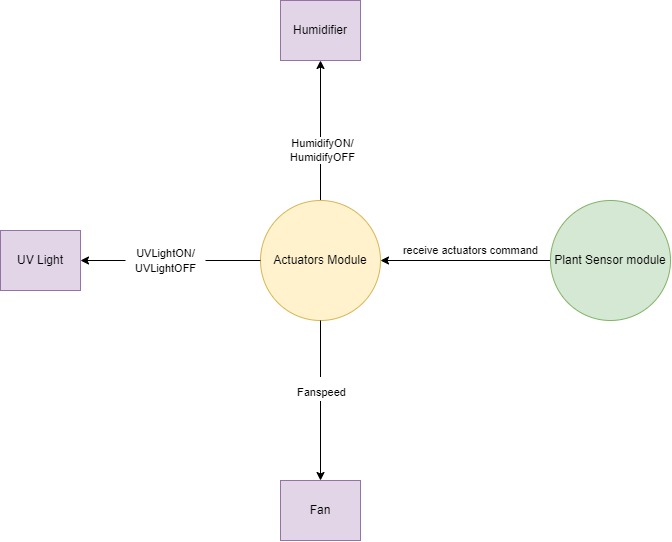


Figure 4 Actuator module context diagram

### Gateway Module

## UML diagram

3. UML diagrams of proposed software solution **(luka)**

## Hardware description

|  |  |
| --- | --- |
| ESP32 Board    Figure 5 ESP32 Board | |
| **Hardware type:** | Microcontroller |
| **Function:** | To control a singular function in a device, by interpreting data it receives from its I/O peripherals using its central processor |
| **Consideration:** | For this project, wireless communication between the automated plant modules is needed, ESP32 makes an excellent choice because it has an integrated Wi-Fi chip, which makes it ideal for prototyping IoT or other devices which require wireless and internet connectivity. The ESP32 is also faster and more powerful than Arduino |

|  |  |
| --- | --- |
| BME280    Figure 6 BME280 Sensor | |
| **Hardware type:** | Sensor |
| **Function:** | To measure Temperature and Humidity |
| **Consideration:** | To make sure the temperature and humidity of a plant’s environment is being measured accurately, the BME280 sensor was chosen because it’s an impressive low-cost sensing solution for measuring humidity with ±3% accuracy and temperature with ±1.0°C accuracy, which makes it perfect choice for this project |

|  |  |
| --- | --- |
| Water Level Sensor    Figure 7 Water level Sensor | |
| **Hardware type:** | Sensor |
| **Function:** | To measure water level in a tank |
| **Consideration:** | The working of the water level sensor is simple and easy to understand. The PCB is made from long conductive plates. When the water reaches a certain level the conductivity between the two plates changes, and by measuring the changes we can measure the water level. This sensor was chosen because it makes it easy and simple to simulate water tank level. |

|  |  |
| --- | --- |
| SG90 180-degree Servo Motor    Figure 8 SG90 180-degree Servo Motor | |
| **Hardware type:** | Actuator |
| **Function:** | To simulate water tank valve |
| **Consideration:** | This servo was chosen to act as a water tank valve, because it is a 180-degree servo, and it can easily be programmed to rotate at any angle between 0-180 degrees, and specific angles can be chosen for open drain and closed drain states. |

|  |  |
| --- | --- |
| Grove Moisture Sensor    Figure 9 Grove Moisture Sensor | |
| **Hardware type:** | Sensor |
| **Function:** | To measure moisture in soil |
| **Consideration:** | To detect soil moisture accurately for plants. The usage for this moisture sensor is simplified. It can be simply inserted into soil and connected to a microcontroller with its Grove cable. It is compatible with 3.3 V and 5 V input voltage which make it more adaptive. You can use this moisture sensor to build loads of smart plant-attending applications. |

|  |  |
| --- | --- |
| Adafruit UV Light Sensor    Figure 10 Adafruit UV Light Sensor | |
| **Hardware type:** | Sensor |
| **Function:** | To measure Ultraviolet light |
| **Consideration:** | Ultraviolet (UV) light is important for plant growth, this UV light sensor was chosen because it uses a UV photodiode, which can detect the 240-370nm range of light (which covers UVB and most of UVA spectrum). This sensor is simple to use, it only does one thing and gives an analog voltage output instead of requiring a complicated I2C setup procedure. This makes it better for simple projects. It also has a 'true' UV sensor instead of a calibrated light-sensor. |

|  |  |  |
| --- | --- | --- |
| LEDs    Figure 11 LEDs | | |
| **Hardware type:** | Actuator | |
| **Function:** | Blue LED | Simulate Humidifier |
| Yellow LED | Simulate Ultraviolet light |
| **Consideration:** | LEDs were used to simulate these actuators because LEDs are simple and easy to demonstrate during a demo. | |

|  |  |
| --- | --- |
| Fan    Figure 12 xsdn12Fan | |
| **Hardware type:** | Actuator |
| **Function:** | To simulate cooling fan |
| **Consideration:** | This miniature fan was chosen to simulate a larger cooling fan that would be used to potentially cool off the indoor climate of a greenhouse when the temperature gets too hot. |

4. Description of chosen hardware and considerations taken into account when making the choices **(sean)**

## Description application protocol

5. Description of the application protocol among the components (What messages should be sent? How often? When? …) **(andre)**

The application that we will be using is Node-Red. With Node-Red we will be sending and/or receiving message protocols to the components. These messages will contain values translated into string messages and can be visualized in the dashboard. These value messages will be sent in moderate. Node-Red will have a function where it will check for messages and if it has not received a message after a certain period of time, the component will go into a timeout and Node-Red will notify the user that it failed to reach the component. The component will be a “plug & play” meaning when it is connected to the server, the component will go into a checklist protocol before going live to the dashboard.

## Communication Protocol

6. Description of the chosen standard communication protocol (e.g. CAN, ESPNow, MQTT, UDP, sockets, … ) among the components and considerations taken into account when making the choices. This communication protocol implements communication for the application protocol described in 5 **(Mitchell)**

The communication between the dashboard and the sensors, is going to be wireless using MQTT protocol in Node-RED, sending and receiving Json info. The sensors will send their measurements to the dashboard, and if they are abnormal, the dashboard will display a message informing the user. We decide to use this protocol because we think it’s the simplest form to make communications that we know.

## State diagrams

This section provides the diagram of our system. Our system is divided into different state machine which shows the message flow.

### Node-Red:

This diagram shows how the connection between the module with the broker which is Node-Red. When a connection has been established, the module will go into idle state and wait for incoming messages.

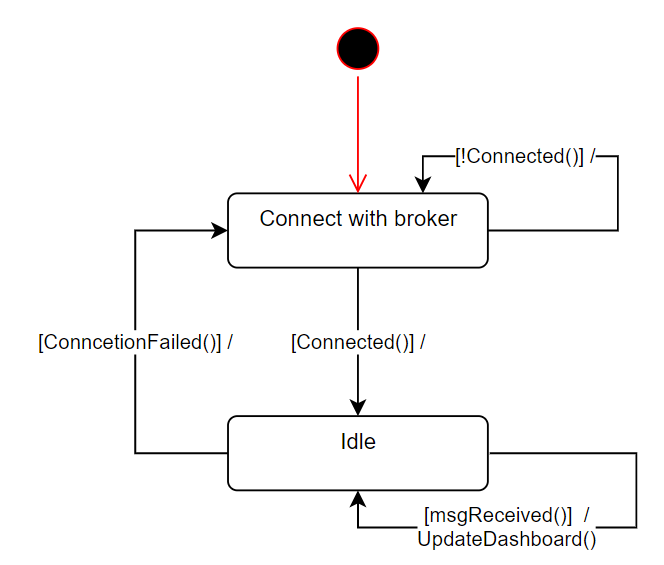


Figure 13: Node-Red state machine

### Gateway Module:

Diagram

Description automatically generated

Figure 14: Gateway Module state machine

### Plant sensor Module:

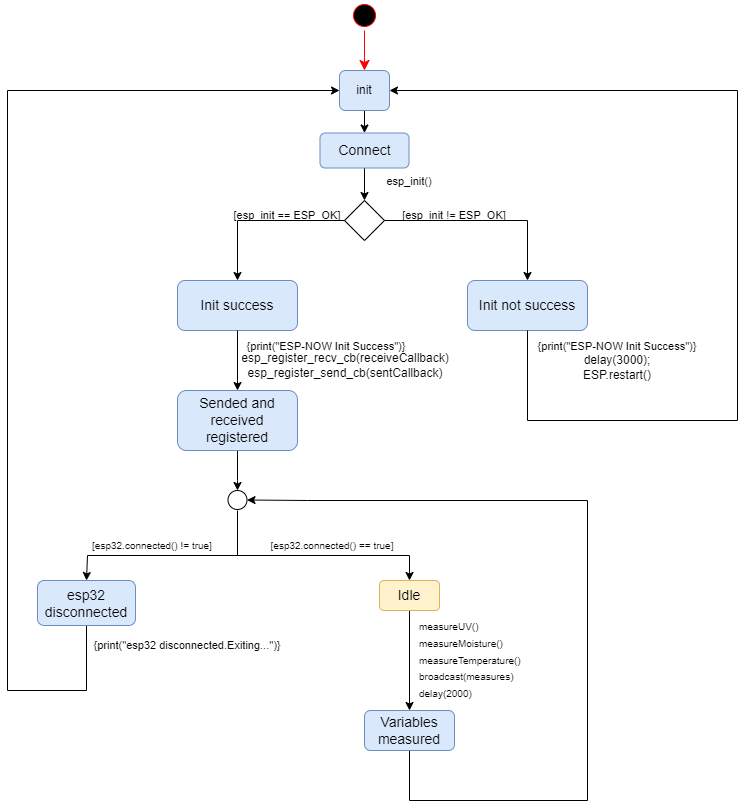


Figure 15: Plant Sensor Module state machine

### Actuator Module:

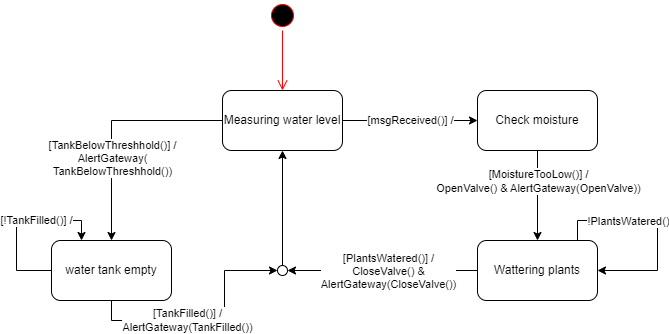
The state machine for the Actuator is given in the figure below. The Actuator Module state receives data and distributes commands to other actuators when needed.

Diagram

Description automatically generated

Figure 16 Actuator Module State Machine

### Water valve Module:



## Proof of Concepts

8. Where applicable description of necessary Proof of Concepts **(andre)**

Our end objective is to make a successful communication between our dashboard and the modules. To clearly define our criteria for success, we will present a prototype of one successful connection to the dashboard with a module mainly an ESP32. With the current knowledge that we have received in the semester, it is possible to be able to program the prototype. For the evaluation of the components, we will be using unit tests to ensure that every situation that is encountered can be avoided. Should the prototype be approved, then moving forward and connecting with the other components will be simple.

## Performance considerations

9. Performance considerations **(luka)**

ESP32:The device of choice to make the project on will be the esp32. The main reason for this choice is the built in Wi-Fi module. This module is extremely beneficial to the team as this eliminates the need for extra system components that take care of the connection.

BME280 (Temperature and humidity sensor):This specific model is choses due to its low cost and surprisingly high accuracy for both humidity and temperature. (+/- 3% and +/- 1%). Another plus for this specific model is the fact that the school already has these in stock.

Grove Moisture Sensor:This sensor measures the moisture of soil. This piece of hardware is perfect for the greenhouse project as one can just directly stick the sensor into the soil and hooked up to the ESP-32. The sensor can be connected to 3.3 and 5V, which is nice for flexibility, as one of the two voltage levels might be too crowded.

Adafruit UV Light Sensor:The main reason this specific sensor was chosen is the fact that can detect all the UVB and most of the UVA spectrum. This is since it makes use of a UV photodiode. The sensor is also easy to use since it returns a simple voltage output, which eliminates the need for a I2C connection.

Water Level Sensor:This sensor is choses due to its easy usage. Its PCB consists of two conductive plates. As the water level reaches a certain level, the conductivity between the plates. changes this makes it easy to simulate the water level.

SG90 180 Degree Servo Motor:This tried and tested servo is in our hardware kit and has been used before in projects, which makes it handy to use again. The servo can rotate 180 degrees which means it can be used to simulate a plethora of things.

LEDs:LEDs are quite straightforward. They are easy to use and can be used to simulate and indicate a large variety of things.

Fan:This fan is used to simulate the cooling down of the greenhouse, and a logical solution, while still being easy to use.

## Flow charts

10. Flow Charts (if they help to clarify design) **(andre)**

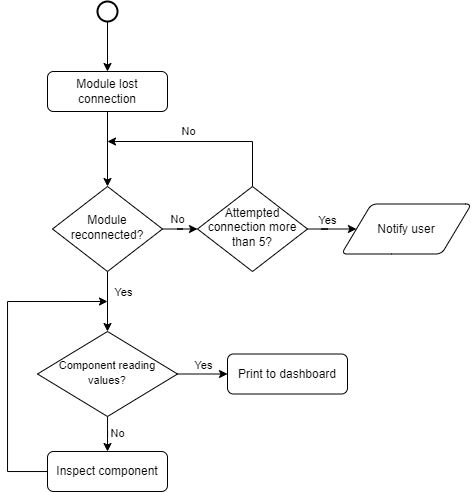
****

Figure 17: Plant module connection Flowchart

The Plant module connection flowchart shows the process of how the module interacts with the dashboard (See figure 9). When the module is disconnected, the system will attempt to establish the connection again and if it has not found after a certain attempt, then it will notify the user. However, if the connection is to be found then it will go into another scan to check if the hardware component is reading its values. If the component detects a value, then it will print it to the dashboard otherwise the component should be inspected.

Diagram

Description automatically generated

Figure 18: Node-Red message handling Flowchart

The Node-Red message handling Flowchart gives a clear view on how messages should be handled. This Flowchart appears in the Node-Red state machine in Figure 13. The Flowchart first listens for messages and when a message is received the ID that is sent with the value/state is being separated from the message. Then the Node-Red looks at how to handle the message and how to display it. If the message is not recognized, e.g. the message has been corrupted, the message is ignored because a new message will be sent within a few seconds.

# Test Plan

The test plan report is a result of the tests according to the above-specified test plan. This is done during the end phase of the project.

## Description

The testing of the requirements from this analysis phase will be conducted by making a test case for each use case scenario. The condition for each scenario will be made, which will be the good condition and the bad condition, and then the test will be executed. The results of the test will be specified in the test plan report

## Specifications

The way the software components tests will be conducted in for this project is with unit testing. Unit testing is a method of testing software where individual software components are isolated and tested for correctness.

For example:

- TEST\_ASSERT\_TRUE / FALSE(condition)

- TEST\_ASSERT\_EQUAL(expected, actual)

# Components

## Sensors

### Temperature & Humidity sensor

Houseplants can only survive in a narrow range of conditions. This is especially true considering their temperature needs. Even if they manage to survive temperatures outside of their optimal range, they won’t be able to thrive and grow.

Temperature needs for houseplants differ across species. However, a generalized ideal range for indoor houseplants is 21-27 °C during the daytime and 18-21 °C at night.

### Moisture sensor

Water is a crucial factor in plant development. That’s why irrigation requires a thoughtful approach, as it should be neither excessive nor insufficient. Soil moisture sensors are extremely useful in determining water levels, considerably facilitating farmers’ efforts, and reducing costs.

A soil sensor enables you to schedule irrigation events more efficiently by either increasing or decreasing their frequency and/or intensity, not to wash off valuable nutrients or, on the contrary, leave the plants thirsty. A remote soil moisture sensor empowers agriculturalists to estimate the water levels without the need to be physically present in the field.

### UV sensor

Ultraviolet (UV) light is extremely important for plant growth. In safe doses, UV light helps plants and crops produce essential plant oils which not only enhance the flavour and smell of fruit but also helps the plants protect themselves from excessive UV exposure, acting as their own natural sunblock.

That is why it is important to know if your plants are getting sufficient UV lighting, which is where the UV sensor comes into play.

### Water Level sensor

Water is important for plants to survive therefore it is necessary to make sure they don’t run out of water supply, in order to be aware if water supply is running low water level sensor is needed in order to get updates if the water tank for example needs to be supplied with more water.

## Actuators

### Servo (valve of water)

A servo motor could be used to act as an automated water valve, that will supply the plant pots with water according to soil conditions.

### Fan (For better temperature levels)

A fan could be used to act as an automated fan that can cool and dehumidify the climate temperature around the plant if necessary, according to climate conditions.

### LED /UV light (For better temperature levels)

A yellow LED could be used to act as an automated UV light, that will supply the necessary UV lighting and temperature for sufficient plant growth and photosynthesis.

### Humidifier (For better humidity levels)

A blue LED could be used to act as an automated humidifier, that will supply the necessary amount of humid for the plant.

# References

*Maintain Temperature for Plants Like a Pro*. (2021, November 9). Smart AC Controller. Retrieved October 17, 2022, from <https://cielowigle.com/blog/temperature-for-plants/>

Cherlinka, V. (2022, August 5). *Soil Moisture Sensor: Innovation For Precision Farming*.  
EARTH OBSERVING SYSTEM. Retrieved October 17, 2022, from <https://eos.com/blog/soil-moisture-sensor/>

*What types of sensors are used in precision agriculture?* (2022, June 30). GeoPard Agriculture. Retrieved October 17, 2022, from <https://geopard.tech/blog/what-are-the-types-of-sensors-used-in-agriculture/>