Indoor farming using HVAC and IoT

Group ID: Aniruddha Jahagirdar, Rachana Nagaraj, and Sowjanya Krishna

Service Computing Department, IAAS, University of Stuttgart
st175081@stud.uni-stuttgart.de st173597@stud.uni-stuttgart.de
st174999@stud.uni-stuttgart.de

Abstract. Smart farming has taken a leap from traditional farming as it brings reliability and predictability at our fingertips. Automation and cloud software systems are tools for smart farming. Smart farming emphasises on acquiring data collected by sensors and make productive use of it. Smart farming employs hardware (IoT) and software (SaaS) to capture the data and give actionable insights to manage all the operations on the farm. The data is organized, accessible all the time that can be monitored remotely and allows us to take necessary actions. Smart farming analyses soil moisture content, indoor room temperature and humidity and light intensity to get knowledge about suitable crops and water requirements for optimization. In this project, different sensors are used to detect real-time physical conditions like temperature, humidity, light intensity and soil moisture content and send the data to Raspberry Pi. Then, the data will be processed by Raspberry Pi and uploaded to the Cloud. Furthermore, the system will automatically maintain ambient conditions by modulating HVAC devices, indoor lighting and water pumps to save energy. The web-based application aims to graphically represent the gathered data to provide better understanding of the facility. With the help of indoor farming using HVAC and IoT, productivity and quality of crops are expected to increase significantly. Additionally, reduction of carbon footprints is also achieved.

Keywords: Smart farming · HVAC · Carbon footprints.

1 System Introduction

1.1 Project Scope

Due to the increase in population and peak country developments, alternative options are being sought for feeding the masses, yet minimizing the land used. Water and land are both classified as finite resources. Thus, farmland need to be shrink in order to provide more land for housing and building. However, farmland is vital to produce oxygen and sustain food supplement all around the world. One of a latest technologies introduced in agriculture field to diminish the land used issue is the vertical farming, which is also an effective way to grow plants [2]. When practicing indoor farming, most of the farmers would like to monitor the farming conditions, and yet limited knowledge on data management have forced

them to investigate plant conditions with naked eyes [3].Therefore, vertical farming monitoring system with Internet of Thing (IoT) is introduced as a platform to collect data and visualize it through a web-based applications. It is a more convenient way to keep track on the vertical farming performance, and contribute to the research and development on agriculture field with real case data analysis. With the help of the system, a reliable environment for farming activities could be formed. Consequently, crops are grown under a controlled surrounding, and get rid of the dangers caused by extreme weather such as droughts and floods [1].

1.2 Project Goals

This project aims at automating of a traditional farming by using HVAC technology and smartly managing the operations. Fundamentally, project is divided into 3 major parts.

1. Different type of sensors will collect data like temperature, humidity, light intensity and soil moisture content.
2. Data will be transmitted to Raspberry Pi. After processing the data, Raspberry Pi will send the sensed data to online AI planner via MQTT.
3. Online AI planner will take the decision accordingly and command actuators to maintain optimum temperature, humidity, light intensity and soil moisture to meet all the set points.
4. Data will displayed on a web based application in order to deliver the information clearly, with the aid of graphs and figures.

Hence this project aims to automate indoor farming with help of modern technology.

2 System Analysis

2.1 Requirements

Following are the requirements of the system:

1. HVAC devices regulating temperature and humidity
2. Controlled artificial lighting
3. Soil moisture monitoring
4. Representation of measured parameters
5. Determination of composition of soil
6. Measurement of Energy consumption

The user requirements can be assigned based on functional and non-functional requirements. The above listed requirements are all functional. Additionally, the requirements are classified based on their priority.

Table 1. User Requirements for Indoor Farming using HVAC and IoT.

Sr. No.	Description	Type	Priority
R01	Maintaining Optimum Temperature and Humidity with HVAC equipment	Functional	Mandatory
R02	Maintaining Optimum Light Intensity with controlling artificial lighting	Functional	Mandatory
R03	Maintaining Soil moisture by controlling supply water valve	Functional	Mandatory
R04	System monitors all processes and sensors in real-time and displays on the GUI. Alarms are popped on GUI if any parameter goes out of range	Functional	Mandatory
R05	Calculation of energy values of actuators and generation of carbon footprint	Functional	Desirable
R06	Generation of random sensed values of Zinc, Magnesium and Nitrogen. Soil health prediction based on these values	-	Future Scope

2.2 Use case Diagram

A use case diagram represents the interaction between the system and its different users. For each user, the use case diagram presents the different use case in which he is involved.The admin is an actor who monitors the overall system. A graphical user interface (GUI) enables the supervisor to efficiently manage the resources in the system, he also checks the HVAC and pumps if any issues occur. This person also has access to redefine the agricultural parameters for different crops with respect to the farmer's demands. The system user is the Farmer who has access to monitor the GUI to evaluate the crop growth and determine his harvest.

User Story:

- As the supervisor of the system I need to check the Energy Consumption.
- As the supervisor of the system I need to check the functionality of the actuators.
- As the supervisor of the system I need to redefine the agricultural information for different crops.
- As the supervisor of the system I want to monitor the GUI.
- As a Farmer I can access the GUI.
- As a Farmer I want to view the agricultural information.
- As a Farmer I need to harvest the crops.

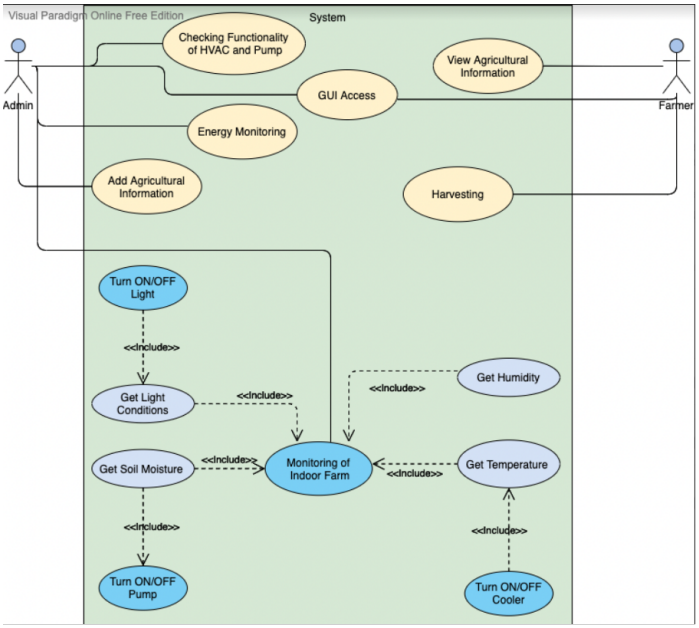

Fig. 1. Use Case Diagram

3 System Architecture Design

The system architecture consists of Raspberry Pi, sensors like humidity, temperature, soil moisture, light intensity sensor and actuators like automatic lighting system to turn on/off or to maintain the light intensity, HVAC devices that deliver the heat, cooling, air filtration, and water pump.

The model is divided into Four layers: User Layer, Reasoning Layer, Ubiquitous Layer, Physical Layer. The first layer is the physical layer, which describes the system's hardware, software, and network environment. The perception layer is the physical layer, which has sensors for sensing and gathering information about the environment and actuators which enables actions to be taken in the environment interfaced to the Raspberry Pi. MQTT protocol is used for the sensors to send data to Raspberry Pi's mosquitto MQTT Broker, which client devices can then receive that data. In the Reasoning/Ubiquitous layer, AI planner is implemented which gives a sequence of action depending on the outcome of the planner, the data is published back to the raspberry Pi for the actuation. The user layer consists of Graphical User Interface (GUI) which provides insight real-time data like temperature, light intensity, soil moisture and humidity.

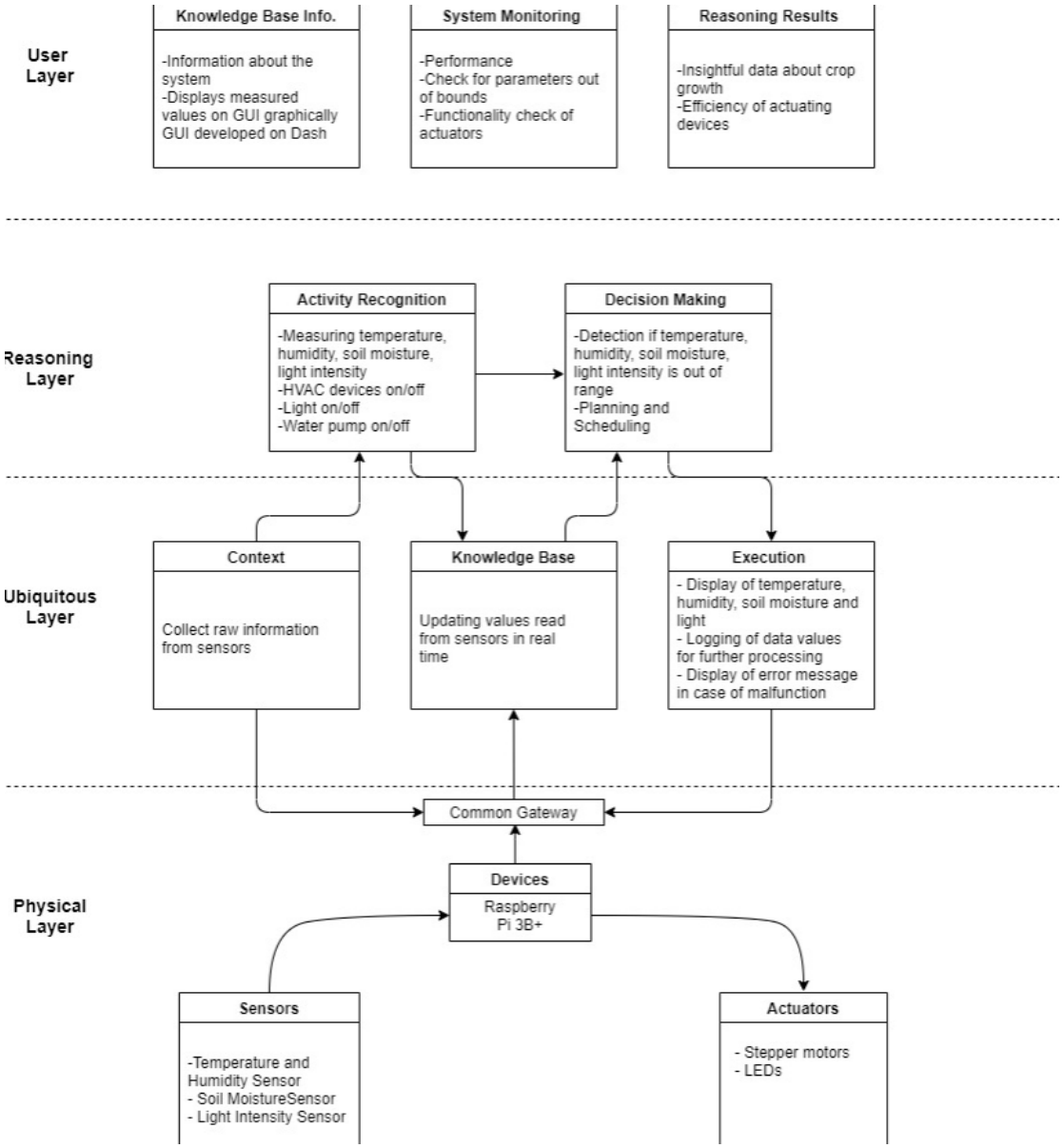

Fig. 2. Architecture Design Of the System

Hardware part consists of:
i) Raspberry Pi
This controls all the activities taking place on the board and acts as an IoT gateway. GPIO pins are used to connect the sensors and actuators on to the board.
Technical specification:

- Quad core 64-bit ARM-Cortex A72 running at 1.5GHz
- 802.11 b/g/n/ac Wireless LAN
- Bluetooth 5.0 with BLE
- 1x SD Card
- USB-C power supply capable of delivering 5V at 3A

A) Humidity Sensor-DHT11
The percentage of water present in the air is termed as humidity. Humidity measurement in smart farming is critical because air moisture may affect the grown of the plants.The change in RH (Relative Humidity) of the surroundings would result in display of values. The DHT sensors are made of two parts, a capacitive humidity sensor and a thermistor.
Technical specification:

- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80 percent accuracy

B) Temperature Sensor-DHT11
This measures the temperature of the room and this data can be used to optimize the temperature for the plants grown.The DHT11 is a basic, ultra low-cost digital temperature, a thermistor is used to measure the surrounding air, and gives out a digital signal on the data pin (no analog input pins needed).
Technical specification:

- Vcc- Power supply 3.5V to 5.5V
- Data-Outputs Temperature through serial Data
- Operating current: 0.3mA (measuring) 60uA (standby)
- Temperature Range: 0°C to 50°C
- Accuracy: ±1°C

C) Soil Moisture sensor(v1.2):
Soil moisture sensor measures the water content in soil. A capacitive moisture sensor works by measuring the changes in capacitance caused by the changes in the dielectric. It does not measure moisture directly (pure water does not conduct electricity well), instead it measures the ions that are dissolved in the moisture These ions and their concentration can be affected by a number of factors, for example adding fertilizer for instance will decrease the resistance of the soil. Capacitive measuring basically measures the dielectric that is formed by the soil and the water is the most important factor that affects the dielectric. Here, It is used to sense the moisture in field and transfer it to raspberry pi in

order to take controlling action of switching water pump ON/OFF.

Technical specification:

- Supports 3-Pin Sensor interface
- Analog output
- Operating Voltage: DC 3.3-5.5V
- Output Voltage: DC 0-3.0V
- Interface: PH2.0-3P

D)Light Intensity Sensor
LDR sensor module is used to detect the intensity of light.When there is light, the resistance of LDR will become low according to the intensity of light. The greater the intensity of light, the lower the resistance of LDR. The sensor has a potentiometer knob that can be adjusted to change the sensitivity of LDR towards light. According to the available data the intensity of the light is adjusted. Operating voltage: 3.3 V - 5 V Output: digital 1/0

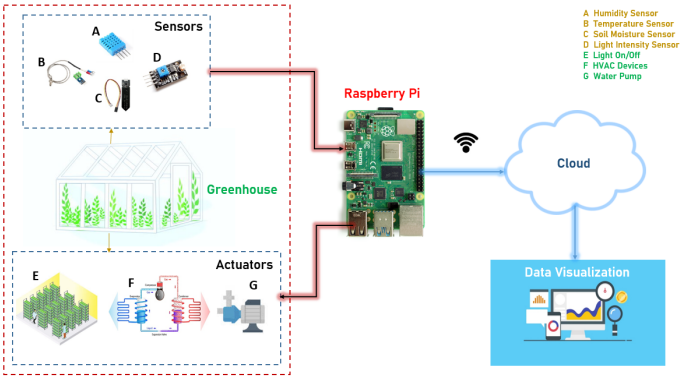

Fig. 3. Block Diagram of Indoor Farming

4 System Implementation

4.1 AI Planning

The planning problem in Artificial Intelligence is about the decision making when trying to achieve some goal. It involves choosing a sequence of actions that will (with a high likelihood) transform the state of the world, step by step, so that it will satisfy the goal.

1. Environment-The indoor farm, Crops, Actions, States

2. Agent-Supervisor

3. Initial State – HVAC devices are OFF, Lights OFF, Seedling of crops

4. Actions- Maintain optimum:
   - Temperature
   - Humidity
   - Soil Moisture
   - Light Intensity
   - Differentiate optimum values for different crops
5. Goals:
   - Turn OFF Light during night and maintain intensity during the day
   - Turn ON cooler when temperature is high
   - Turn ON heater when temperature is low
   - Turn ON motor when soil moisture is low and vice versa
   - Open vents when ventilation is required

Given this information the AI planner will find a plan consisting of a sequence of action that has to be followed in order to determine the goal. Depending on the type of the crop the system send the correct optimum values and triggers a python planner.py which generates the plan for each sensor by giving a post request to the server. The input to the python script are problem and domain pddl file for each sensor conditions in the system. When the plan is generated, we analyse the plan file and get the necessary action items. Depending on the action items the data is published back to the Raspberry Pi and the actuation takes place thereby updating the status onto the User Interface as well.

4.2 PDDL

Planning Domain Definition Language (PDDL) is a family of languages which allow us to define a planning problem. It is a standard encoding language for "classical" planning tasks. Planning tasks specified in PDDL are separated into two files:

1. A domain file for predicates and actions.
2. A problem file for objects, initial state and goal specification.

Domain File:
A domain file, typically named domain.pddl, has the following sections: The name of the domain, a list of predicates and each predicate's variables (recall that a predicate represents a single property of a state).Predicated are properties of objects that we are interested in; can be true or false.,a list of actions; for each action, its parameters, preconditions, and effects may be stated. The parameters define the variables for the state mentioned in predicates. The preconditions are the pre-requisites which need to be satisfied in order to achieve the effects. The effects are the results obtained.

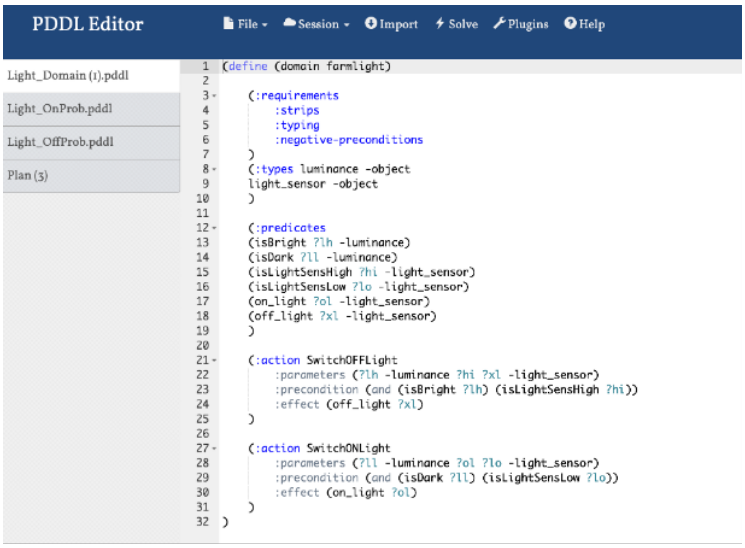

Fig. 4. Domain File for Light Sensor

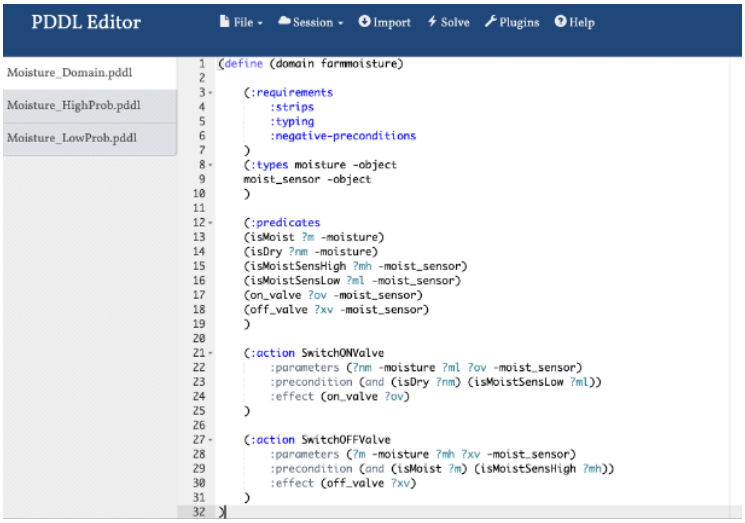

Fig. 5. Domain File for Moisture Sensor

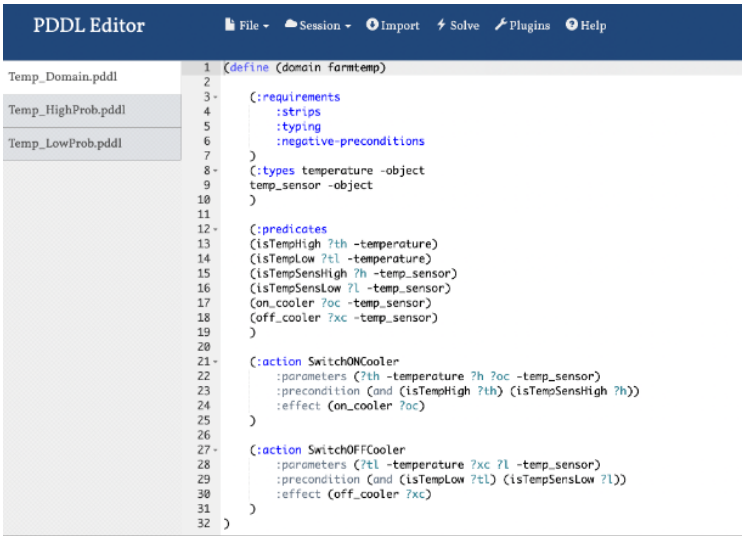

Fig. 6. Domain File for Temerature Sensor

Problem File: A problem file consists of objects, initial state and goal specification. In our case, the problem which we deal with is to sense the environment and trigger the right actuator to provide optimum conditions for the plants.

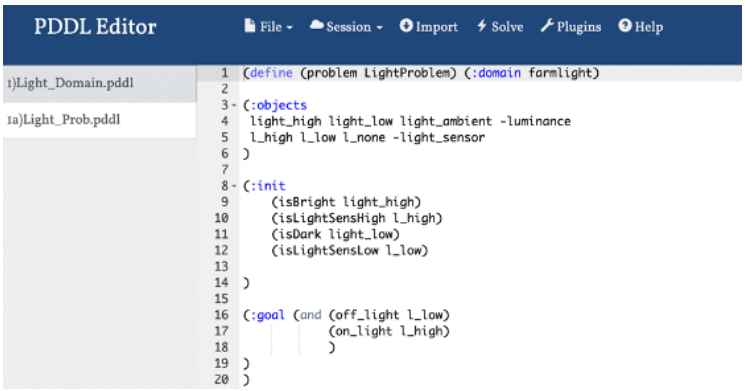

Fig. 7. Problem File for Light Sensor

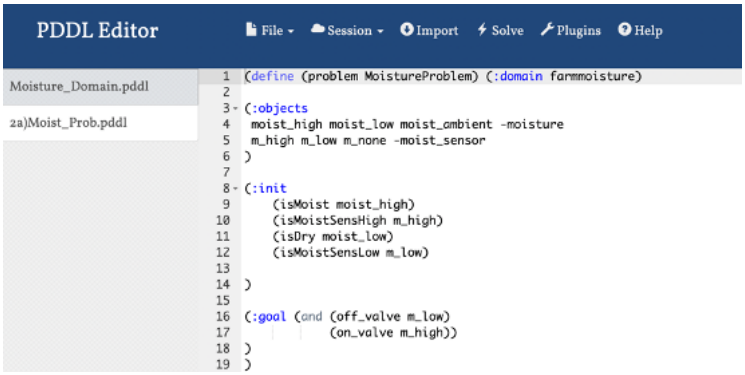

Fig. 8. Problem File for Moisture Sensor

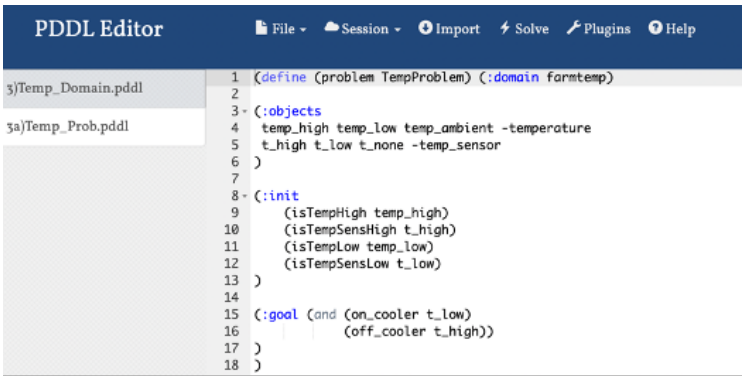

Fig. 9. Problem File for Temerature Sensor

Result of PDDL

Fig. 10. Plan for Light Sensor

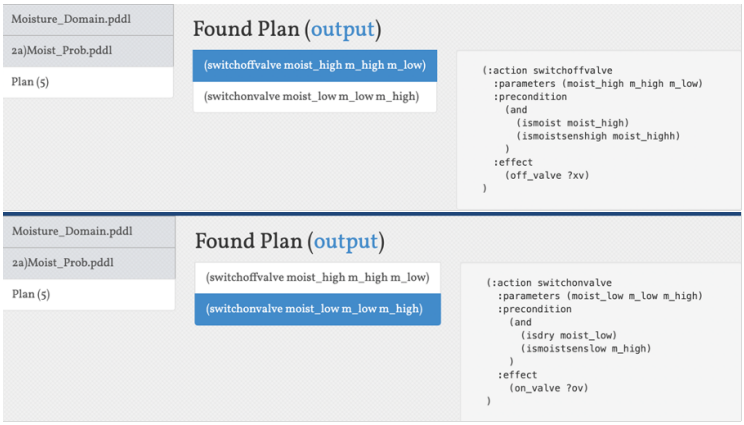

Fig. 11. Plan File for Moisture Sensor

Fig. 12. Plan File for Temerature Sensor

4.3 Working

The proposed smart indoor vertical farming system consists of sensors, actuators, and Raspberry Pi. Raspberry Pi Model is a single board computer, it serves as a local control device for the actuators and sensors. Three main purposes of Raspberry Pi are: a) collect data from the sensors, b) transmission and reception of the data over the MQTT and c) actuation. We have interfaced a Humidity sensor, Temperature sensor, Soil Moisture sensor, Light intensity sensor, LED, and two stepper motors. The Humidity sensor measures the humidity in its environment, Temperature sensor that measures the temperature of the environment, Soil moisture sensor measures the water content in the soil and can be used to estimate the amount of stored water in the soil horizon; Light intensity sensor that indicate the intensity of artificial light. Raspberry Pi also receives the actuations to be performed on the LEDs which are used to demonstrates the functionalities of artificial light. Two stepper motors are used where one is used to exhibit the behaviour of a water pump valve and another for HVAC devices (VAV box).

Each Vertical Farming stack consists of one environmental node, responsible for measuring environmental parameters: temperature, humidity, soil moisture, and light intensity. Each layer has pre-set optimum values for each sensor depending on the crop type. The Raspberry Pi collects real-time data from these sensors and checks for the preconditions defined in the python program.

To read all the sensors on a Raspberry Pi simultaneously and in real time and to control several LEDs and motors at the same time, but independently of each other we use a concept known as multithreading. Multiple threads within a process share the same data space with the main thread and can therefore share information or communicate with each other more easily than if they were separate processes. Multithreading is a light-weight process and they do not require much memory overhead.

We have implemented AI planning for the actuation needed during different environmental conditions. Raspberry Pi sends sensor data to the Mosquitto server using MQTT, in the server side we are comparing the sensor data with optimum values needed for different crops. When the temperature sensor reads above 26°C which means the temperature is High, then PDDL scripts will give us the plan for actuation. Cooler will be turned on inorder to reduce the temperature, so the stepper motor will be actuated to demonstrate the Turn ON of cooler. Once the temperature reads low i.e, below 22°C, PDDL gives us the plan to turned off the coolers. Similarly, when Light intensity sensor reads high i.e, when there is enough light during the day, the PDDL script will give use the actuation plan to Turn ON the LEDs and vicevera. The same is implemented for the soil moisture sensor which actuates the motors when the sensor reads low water content in the soil.

4.4 Automation

Smart Indoor Farming: The proposed system aims at reducing the human interference and thus also reduces the error and the wastage of resources. Since all the necessary actions are automated, the farmer can take the necessary actions from a remote location.A fully-automated system that grows produce in a space-efficient controlled environment. Vertical farms replace the sun with LED lighting, which helps to maximize yield per acre by stacking plants on top of each other and giving the plants the optimal amount of light 24 hours per day. The yield is achieved by growing plants at 24 °C, and by using artificial intelligence (AI) to ensure the environment is optimal for each specific plant, including the day and night temperatures, amount of water and light needed. Our sensor system ensures each plant gets exactly the amount of temperature, humidity, water and light which varies for different crops. Energy consumption of the whole system is very crucial and needs to be monitored regularly, this is calculated using the energy consumed by all the sensors and actuations this is then converted into percentage which is displayed on the GUI for the supervisor to observe the usage.

4.5 Indirect Communication

As part of the Message Queuing Telemetry Transport (MQTT) network protocol, messages can be sent between devices using a lightweight, publish-subscribe network. We have used Mosquitto MQTT broker as the communication medium between the Raspberry Pi and another system connected on the same network. Raspberry Pi, first publishes the sensor data, which is then captured by another system via Mosquitto MQTT broker. The system subscribes to a topic and retrieves the data. After running the AI planner, the obtained plan for each sensor is published. The Raspberry Pi now subscribes to the topic and actuation is carried out. Mosquitto MQTT broker was used for the ease of its service.

4.6 Graphical User Interface

All the values from the Raspberry Pi are gathered and displayed on the GUI so that it can be remotely monitored. The GUI is implemented using Dash which is based on HTML and Python. Developed as an open-source library by Plotly, the Python framework Dash is built on top of Flask, Plotly.js, and React.js. Dash allows the building of interactive web applications in pure Python and is particularly suited for sharing insights gained from data. All the sensor data is recorded in real time in a CSV file which is used for plotting the graph. The real time changes in the temperature, light humidity and soil moisture is reflected on the graph and also the light energy and pump energy is documented for evaluating the energy savings. The supervisor can also see these changes in a graphical format. The GUI is the client side of the web server. When we run the indoorfarmdashboard.py script, the GUI appears at localhost: http://127.0.0.1:8050/ in the browser. The home page of the UI(fig:12) gives a brief description about the project and the crops that are harvested. On the main page of the UI-Plant 1(fig:13), we display all the parameters and the same is done for other crops.

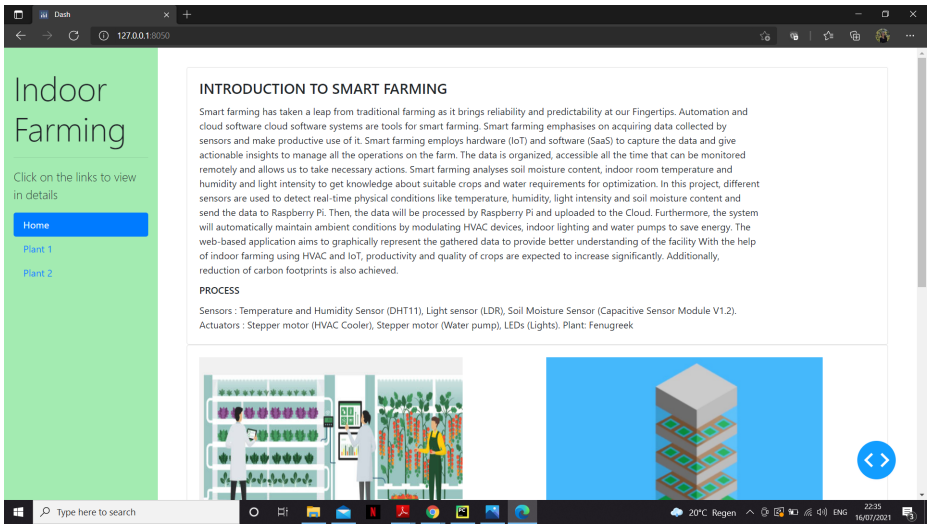

Fig. 13. Home Page of GUI

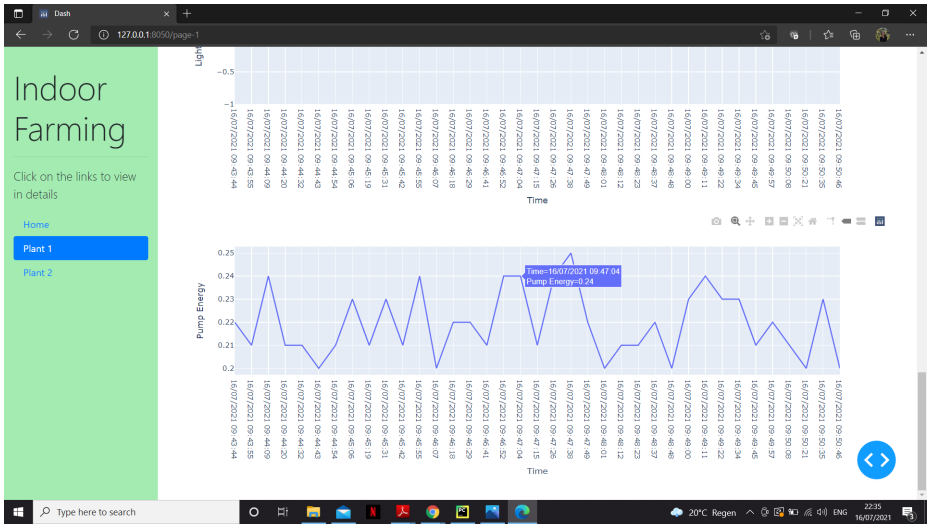

Fig. 14. GUI of Plant 1

4.7 Future Scope

The prototype shows room for improvement in terms of reliability, usability, energy consumption and sensor measurements. To increase automation of the system we will develop a service for automated growth tracking. Further, we plan to build a database of plants with corresponding growth plans, depending on the age, type of the plant. Finally we aim to explore dynamic plan building based on environmental factors and plant type. Our vision is to create A.I., machine vision, and data driven technologies that improve yield, quality and predictability of supply. To redefine farming via vertical farming, by employing Machine Learning and robotics.

4.8 Conclusion

There are many challenges to overcome before fears of a worldwide food shortage can be allayed, including rising temperatures and more frequent droughts caused by global warming. These obstacles are making traditional farming methods increasingly inefficient and unpredictable. Traditional farming has also been hit hard by the COVID-19 pandemic. According to the FAO, border closures, quarantines and disruptions to supply chains are limiting some people's access to food, especially in countries hit hard by the virus or already affected by high levels of food insecurity. In this project we proposed a modular indoor

vertical farming system based on infrastructure provided by IoT framework. Vertical farming is sustainable technology of the future. It reduces use of harmful chemicals and unnecessary water consumption. It ensures healthy and fresh food independent of climate or environmental factors. With help of IoT these systems can be integrated to optimize food production and reduce waist and energy consumption. Software and hardware infrastructure for IoT is are essential components to achieve this goal.

References

1. A. Sarkar, M. Majumder: Journal of advanced agricultural technologies. 2 pp. 98–105 (2015)
2. A. Ĉuriü, N. Bogdanoviü, V. Petroviü, M. Mihajiloviü, G Bogdanoviü: 9th international quality conference. Kragujevac, pp. 127–134 (2015)
3. T. H. Gore, M. P. Kote, P. B. Varpe, M. T. Jagtap: International journal of advanced research in computer science and software engineering. 6 pp. 138–141 (2016)