IndoorPlanterator+

Christian James Lee (2232469)

Dennis Kanatselis (2232694)

420-N55: IoT: Design and Prototyping of Connected Devices Section 0002 Fall 2024
Kawser Wazed Nafi, Email: knafi@champlaincollege.qc.ca

INTRODUCTION

The topic of the project is:

1) Indoor Plant Smart Maintenance

The *INDOORPLANTERATOR+* is a project that involves taking care of indoor plants in automated system using watering and lighting control by taking a look at the environment of the plants through the use of many sensors connected to a Raspberry Pi 4 controlled remotely through VNC and RESTful API using primarily the Python programming language for the logic of it all.

PROJECTMOTIVATION

These are the motivations for the project, we are going to answer these questions relating to it:

- Does it save time for plant owners due to the automation?
- Does it prevent plant deaths from under or over-watering with the use of moisture sensors?
- Does it help grow healthier plants by more accurately figuring out light levels and environmental differences?
- How much water do plants need and when?
- What are the light requirements for optimal plant growth?

The system is going to show real-time information on how the plant is doing using the sensors relating to soil moisture, environmental temperature, humidity, and light levels.

TIMELINE

Week 1: Planning

Week 2: Getting all the equipment for the project.

Week 3: Starting the software prototype in Flask

Week 4: Continuing work on the software

Week 5: Start to make and put the hardware together

Week 6: Continue on the hardware part

(Hopefully have it ready early December)

Week 7: Final Testing, tweaking, and submission

SOLUTIONDESIGN

Sensors:

- Soil Moisture
- Light
- Humidity
- Temperature

Objects:

- Water Valve
- IOT Miscellaneous (Lights, Cables)
- Lights

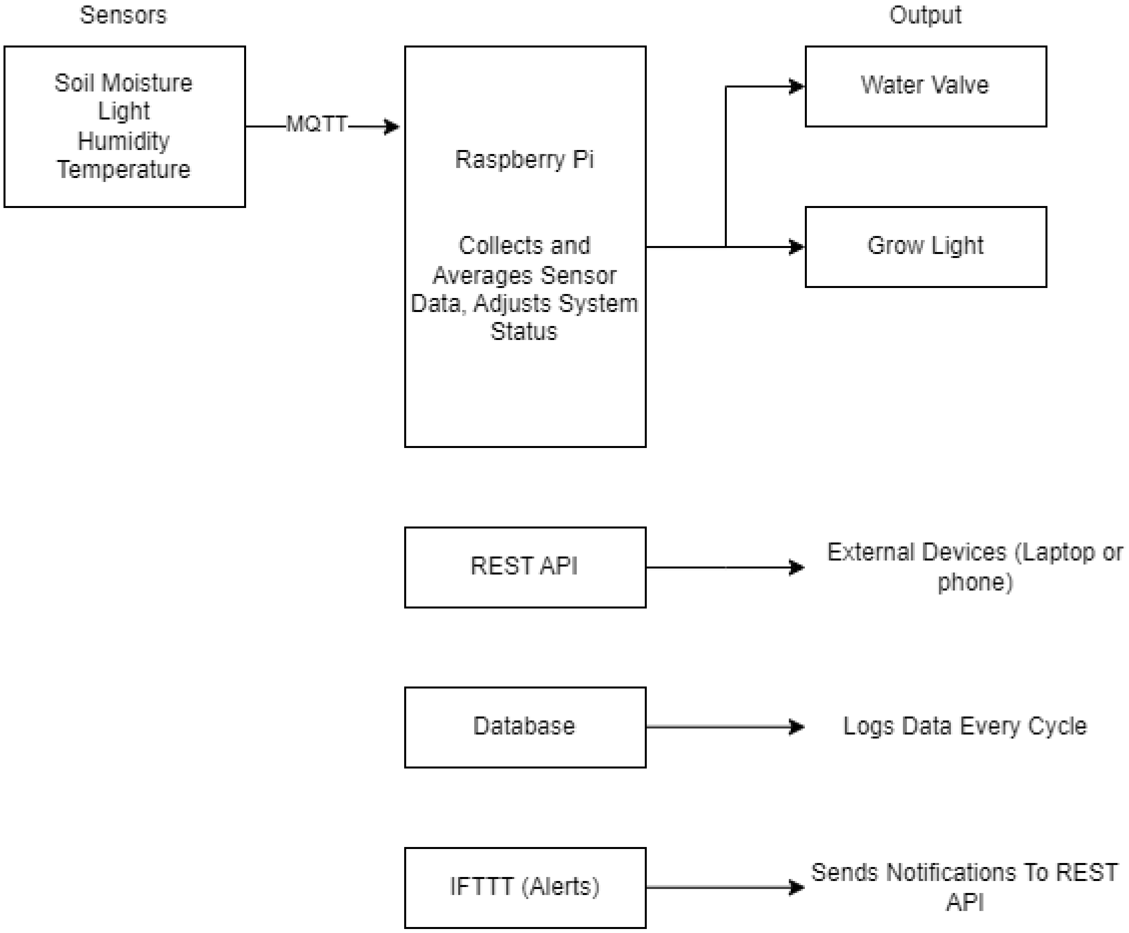
Technology:

- Raspberry Pi 4
- MQTT
- IFTTT
- RESTful API
- Python

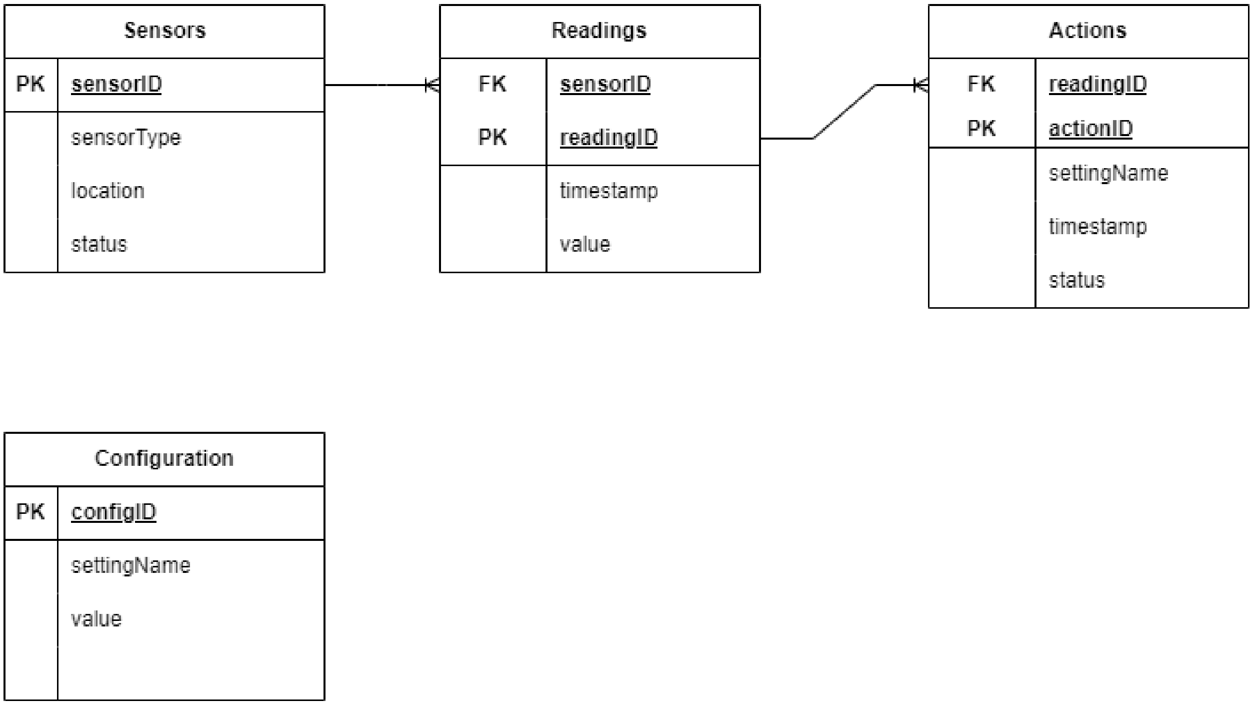
Data Flow:

1. Sensors send data to Raspberry Pi every 30 seconds
2. It collects the data for maybe 5 cycles (undecided) and averages out the results.
3. Based on the collected average, it adjusts the "status" of the system.
4. MQTT for communication between sensors and Pi
5. IFTTT to trigger notifications like if the soil moisture is too low for example.
6. At every cycle, the data is sent to the database, and each average cycle is marked. It will be accessible by RESTful API through pure or formatted JSON.

Solution Design Diagram:



DatabaseDiagram



Sensors	
PK	sensorID
	sensorType
	location
	status

Readings	
FK	sensorID
PK	readingID
	timestamp
	value

Actions	
FK	readingID
PK	actionID
	settingName
	timestamp
	status

Configuration	
PK	configID
	settingName
	value

RESTfulService

From https://planterator.net

GET:

- /api/v1/sensors
- /api/v1/readings
- /api/v1/sensors/{sensor_id}/readings
- /api/v1/alerts
- /api/v1/status
- /api/v1/settings

POST:

- /api/v1/water
- /api/v1/light
- /api/v1/alert/acknowledge
- /api/v1/settings

PUT:

- /api/v1/plants/{plant_id}
- /api/v1/sensors/{sensor_id}

DELETE:

- /api/v1/plants/{plant_id}
- /api/v1/sensors/{sensor_id}

IMPLEMENTATION

The setup will first start with the hardware. All the hardware components will be integrated be within a controlled environment for the most accuracy on monitoring the plant's conditions.

The Raspberry Pi 4 will act like a central hub controller for the sensors, lighting, and water valve system. Additionally, any data collected from the system will be collected to the microSD card stored inside the device.

The data it will collect will come from the sensors used in the project, specifically, soil moisture, light, humidity, and temperature.

LEDS or lights of some sort, able to be activated on command will be used, I am not sure as to what specifically for now but those will be important especially at night where there won't be enough light for a plant to grow properly, especially in an artificial environment like this.

There will be a practically infinite supply of water coming from the house but the amount that is going to reach the plant will be controlled by the water valve which automatically opens up if it detects if there is not enough moisture in the water.

The humidity sensor will be a DHT22 and will be implemented accordingly to work alongside everything else.

To connect to the Raspberry Pi 4 from another computer, VNC will be used, it could be unstable at times but it gets the job done. Preferably the python code will be done and tested on something other than the Pi in Visual Studio Code whenever possible because I believe coding on the Pi will slow down the project time significantly because of how much more tedious it will be.

The other part would be the software and API integration.

Using Python Flask, most of the RESTful requests should be fully implemented with an html menu for some pages. There will be some pages monitoring sensor data for light, moisture, temperature, and humidity connected to the systems and ones that control the water flow, lights toggle, and potentially alerts.

SQLite is what will be considered for the database to store old sensor data that is not current but will be useful.