National Institute of Technology Delhi

National Institute of Technology Delhi

Intelligent Heating System

Presented By:

Tantreshwar Kumar 171220050

ACKNOWLEDGEMENT

I would like to express my gratitude and appreciation to all those who gave me the possibility to complete this report. I consider it a pleasant duty to express my gratitude and thanks to Dr. Rikmantara Basu, Assistant Professor Electronics and Communication Engineering, NIT Delhi.

Through this project I learnt a lot about programming skills and about electronic circuits.

My completion of this project could not have been accomplished without the support of my parents, who always motivated me throughout my journey. I would also like to thank all my friends and my college NIT,Delhi for equipping me with knowledge and making me aware about all recent software tools.

Thanking You
Tantreshwar Kumar

INDEX

Problem Statement

The main problem is to automate the process of heating water. During winters very often we forget to switch off the heater after sufficient heating. Then to adjust water temperature we mix cold water. This leads to wastage of water as well as electricity. It is also important to note that during winters we tend to become lazy, so it is convenient if a heating system operates without going to bathroom.

The best way to handle it to use the concept of Internet of Things and appropriate prediction algorithm. So, the main idea is that whenever a user goes to bathroom, he will find water heated perfectly to the temperature he likes.

.Additionally, to use electricity judiciously we can devise an electronic system which can decide between solar and mains power.

Components Used

Hardware Components:

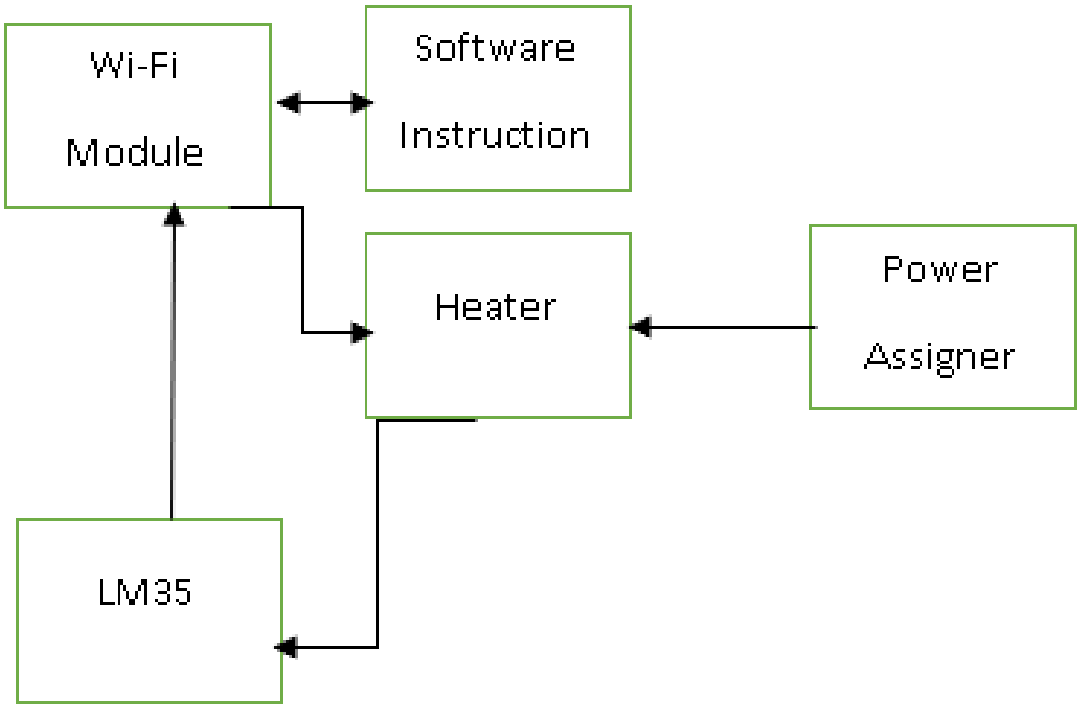
- Wi-Fi Module
- Breadboard
- Relays (Can be replaced with appropriate power electronics)
- Basic Digital Gates
- Comparators
- AC-DC converter
- LM35 sensor
- Heating elements

Software:

- MySQL
- Python

Working Methodology

Overall Block Diagram:



Power Assigner: Decides which power type is suitable. It chooses between solar power and mains power

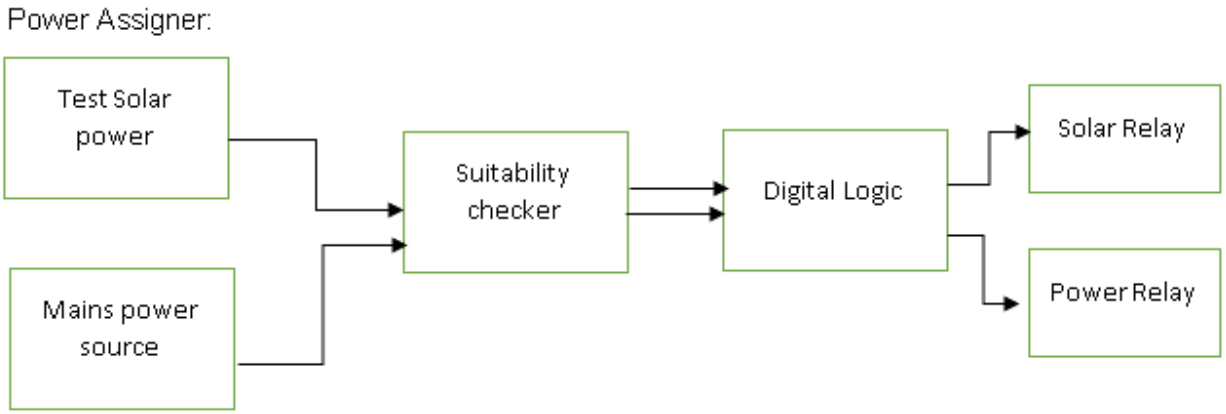
Heater: Device for heating

Wi-Fi Module: Device that takes software instructions command. These commands are given over internet

Software Instruction: Software program to control operations

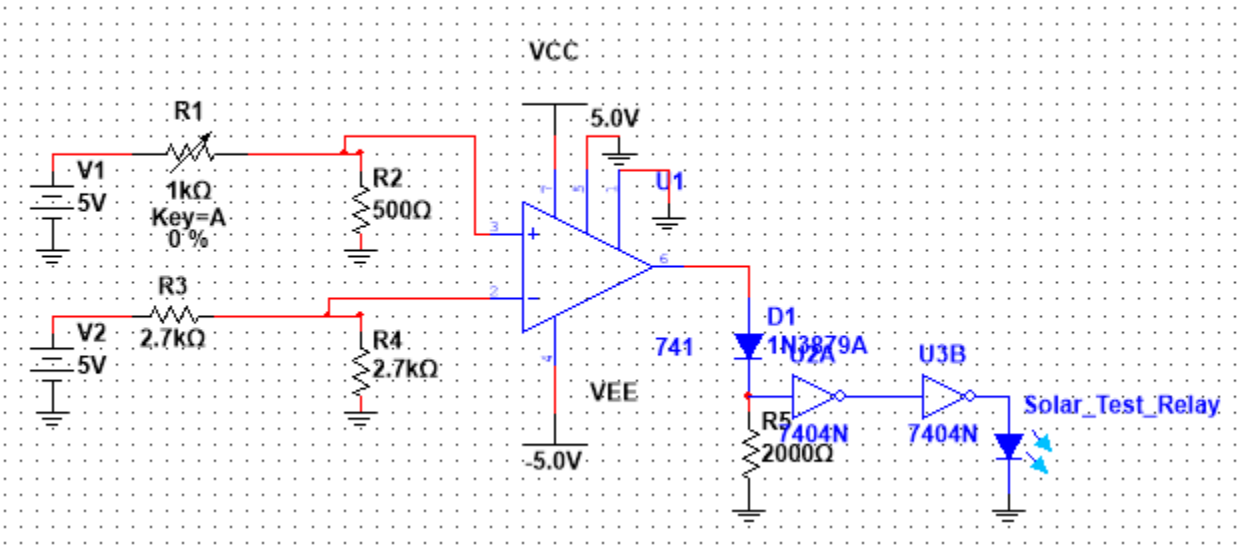
LM35: Temperature measuring sensor

Power Assigner



Test Solar Power:

This part tests whether sun is shining or not. It involves LDR and comparator operations

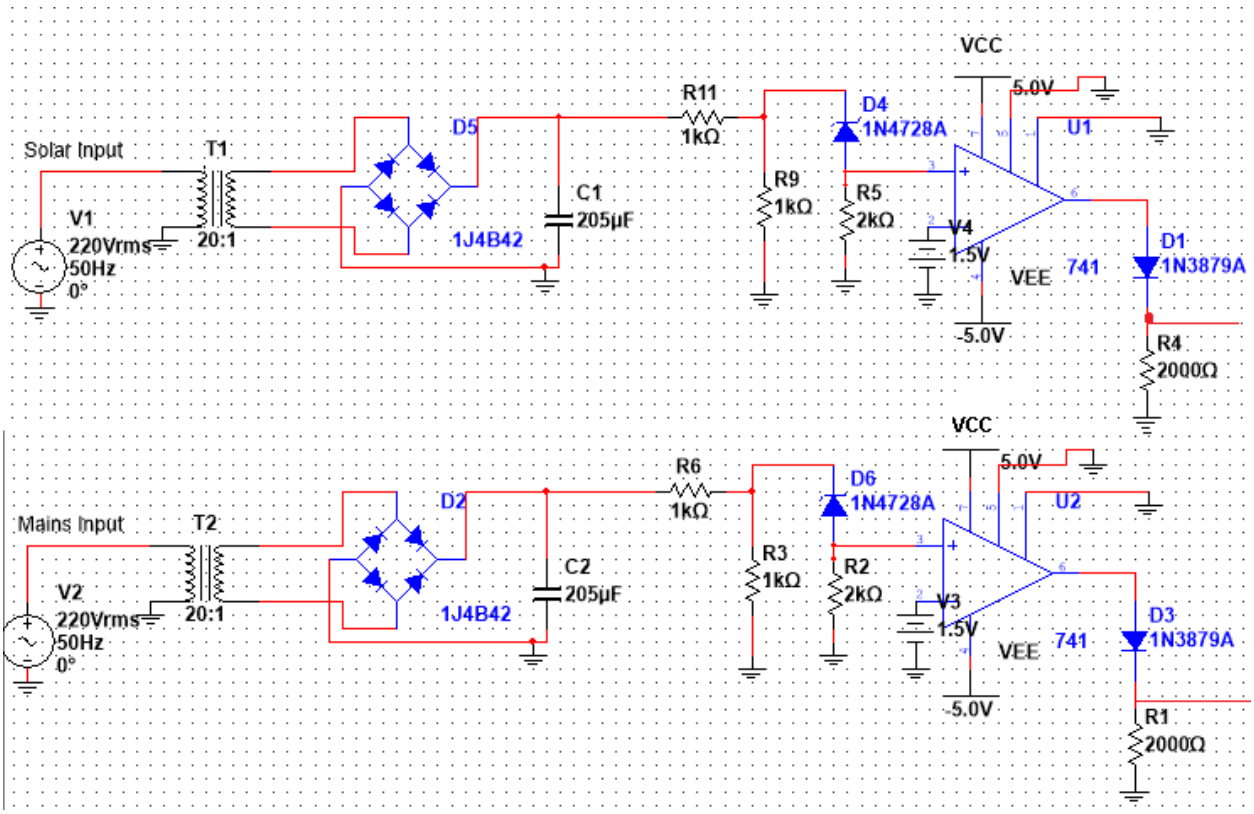


Working:

The potentiometer is like LDR. During good sunshine the LDR will have zero resistance and will switch on. Comparator compares it with 2.5V if low then the diode D1 will get reverse biased else it will allow to voltage to bypass. The 2 NOT gates are connected to convert the high voltage to 5V and low voltage to zero volts

Suitability Checker:

Sometimes low voltage cases are reported this can harm the device. So by knowing minimum voltage requirement of heater we can decide whether the power is suitable or not. Suitability checker gives suitability status through logic 1 or logic 0. The first arrow is suitability of solar second is for mains power.



The first output line is for solar status. Next one is for mains status.

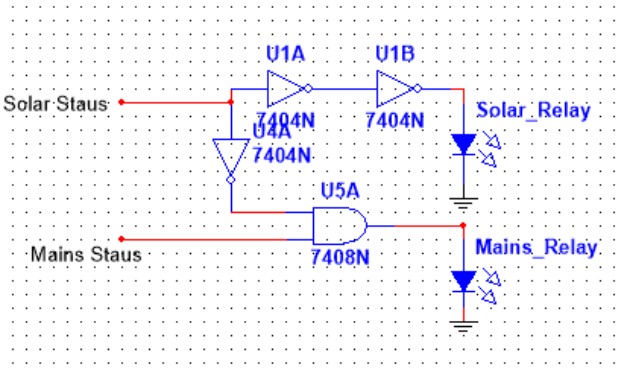
Working:

Firstly, the voltage is stepped down. Then it goes through bridge rectifier and capacitive filter. Now if voltage is high then zenner will breakdown and the voltage is compared with 1.5V. If it is high then comparator gives high if low then it gives low.

To realise it mathematically we have to first know the heater minimum voltage requirement. As per that we have to select zenner diode and dc voltage at inverting terminal of comparator

Digital Logic:

On basis of suitability status, digital logic gives final instruction to output relays.



From suitability checker it takes solar status as input one and another for main status. Logic of this circuit is that:

1. If both solar and mains power is suitable: Take solar power
2. If solar is not but mains is: Take mains power
3. If both are not: Take none
4. If mains is not but solar is: Take solar power

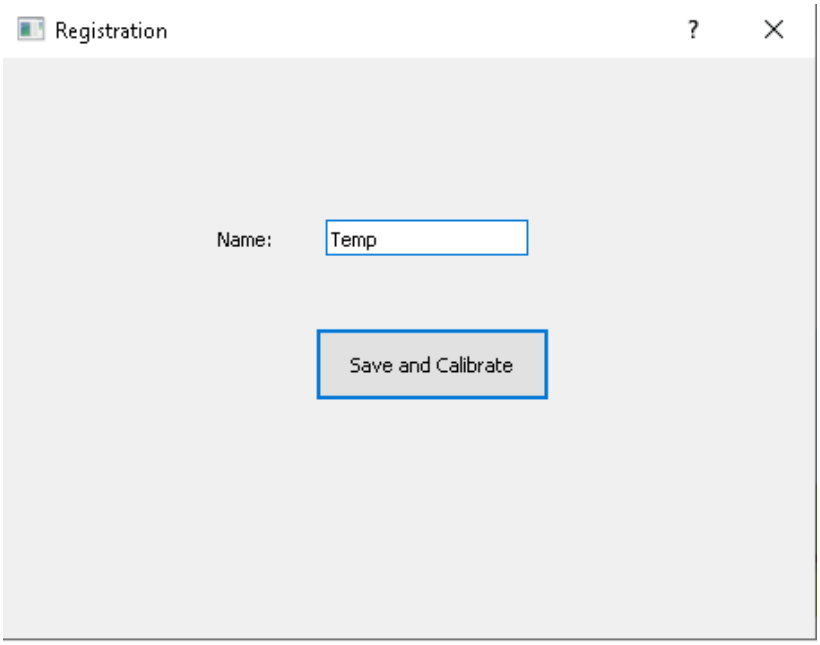
Software Front End

Main Window:

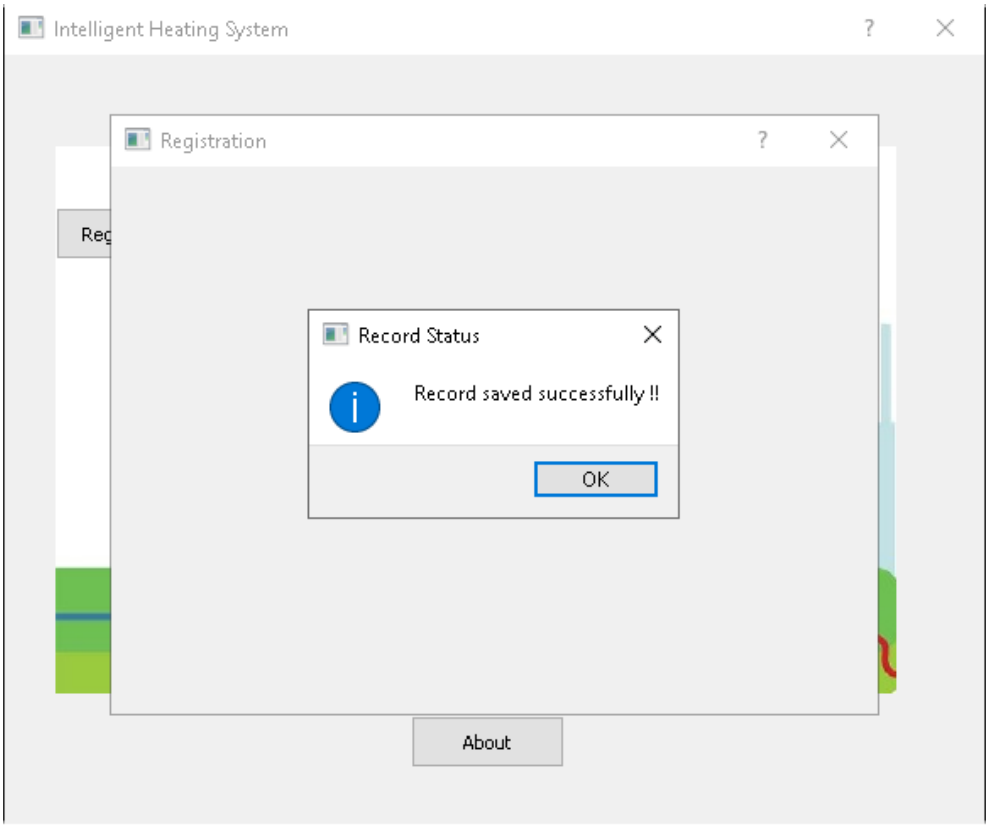
This window is the main window which guides users through the different options available. The functions of all window is described below:

Registration:

Here for the first-time user entry is saved. For the first-time he/she calibrates the reading

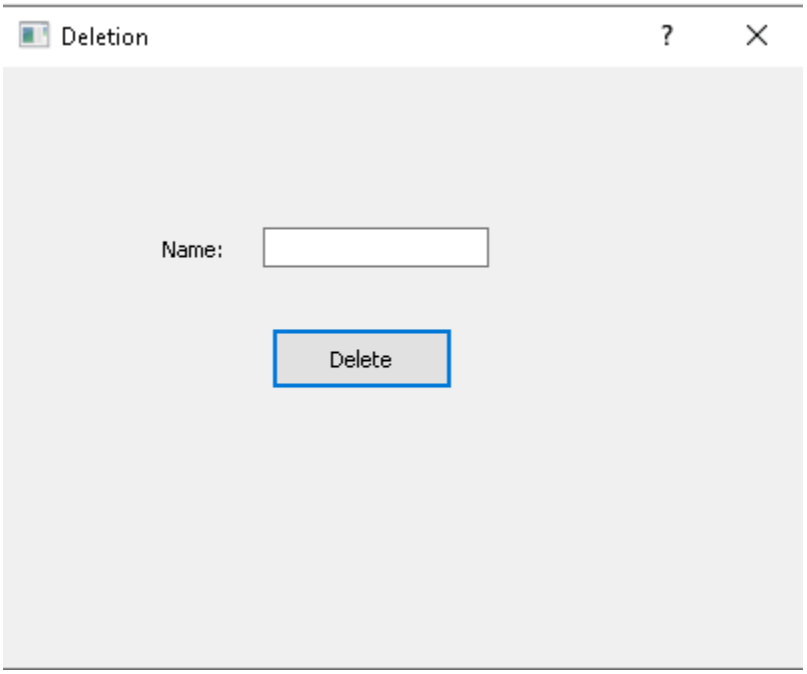


After user types his name and clicks save and calibrate we get:

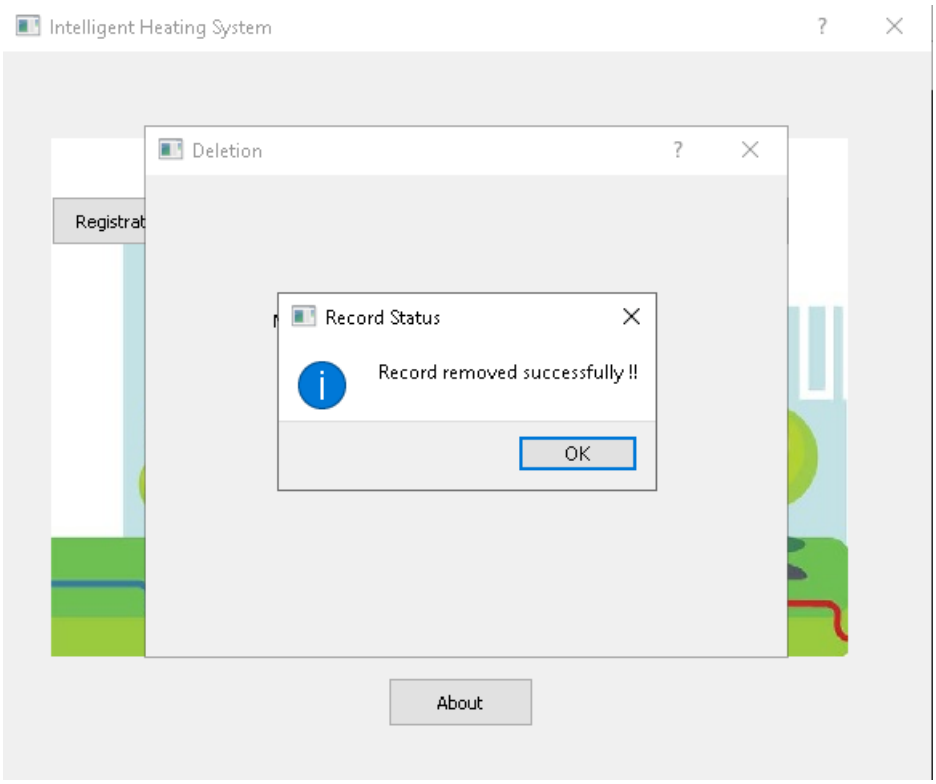


Deletion:

Will remove users record.

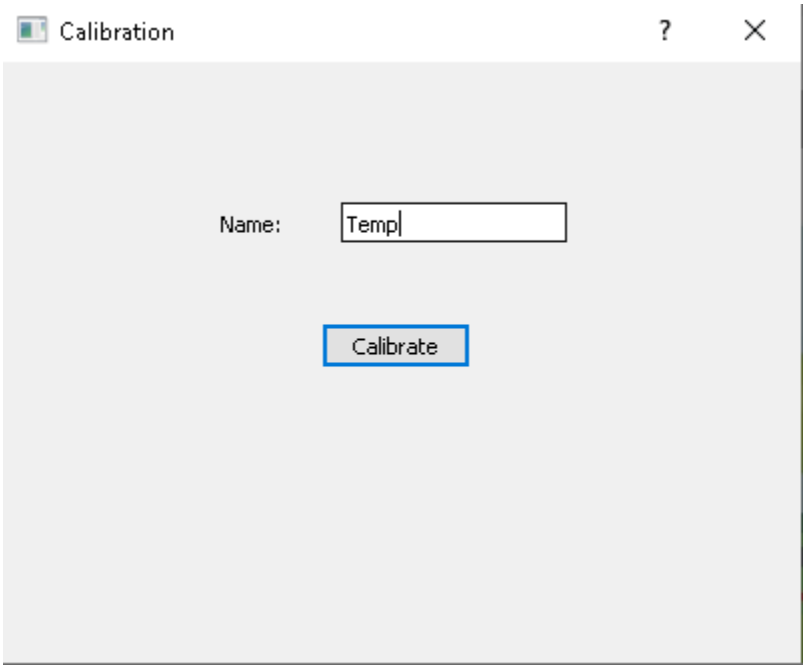


After typing name and clicking delete:

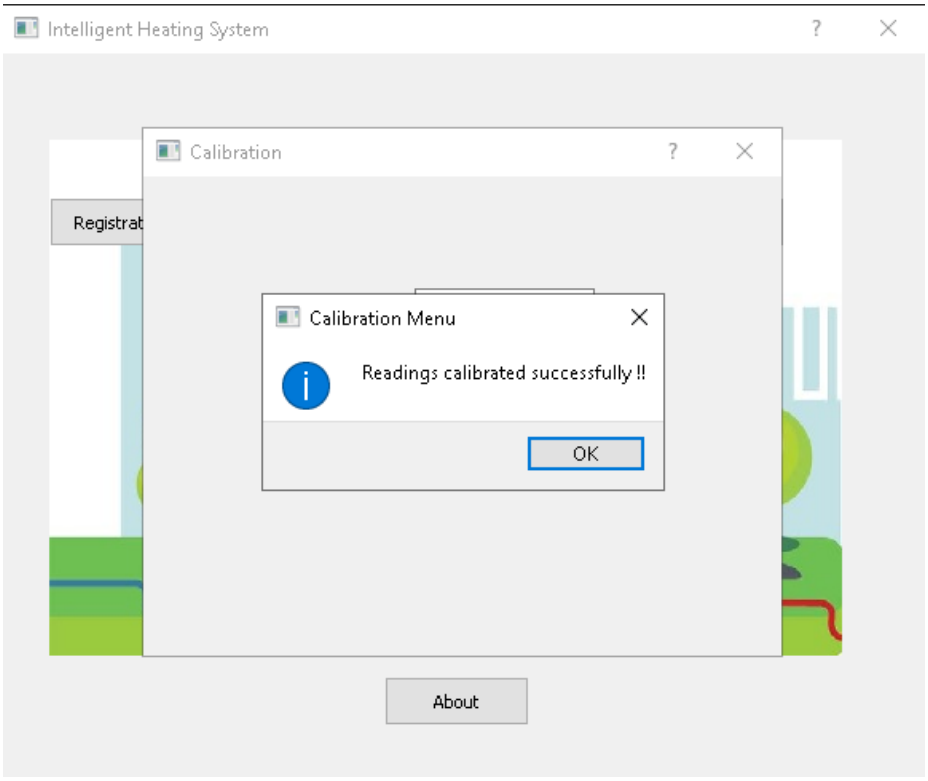


Calibration:

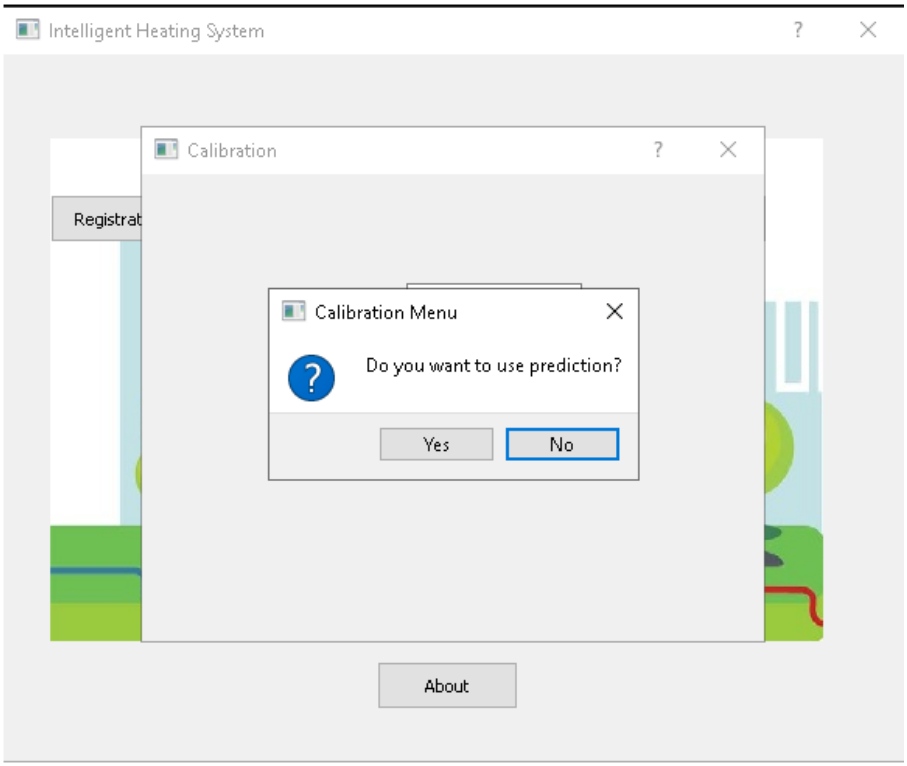
Here user if he is unsatisfied can calibrate once again.



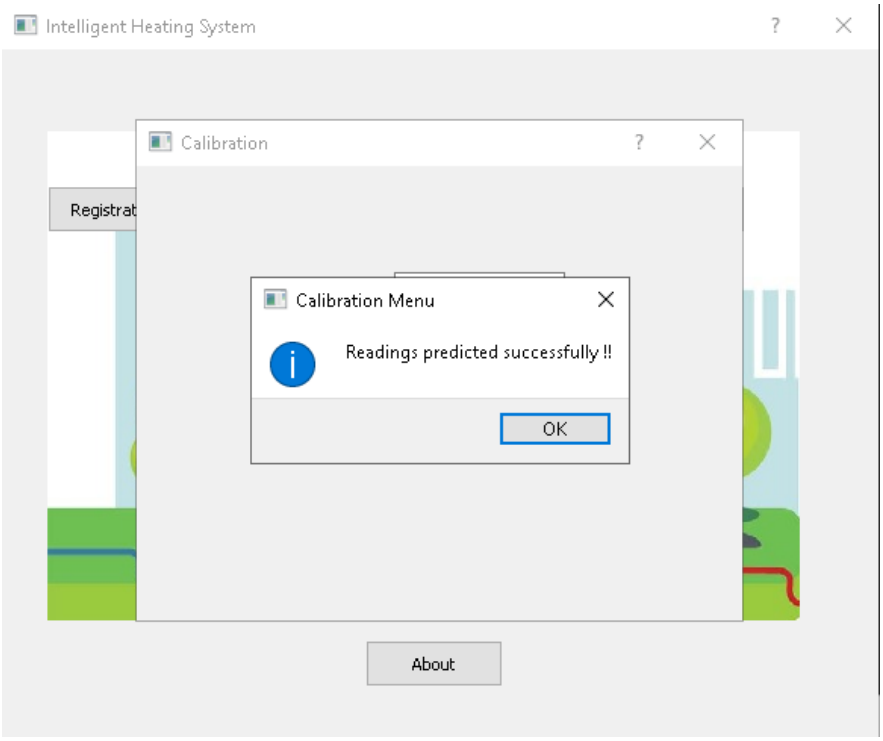
After typing name and clicking calibrate:



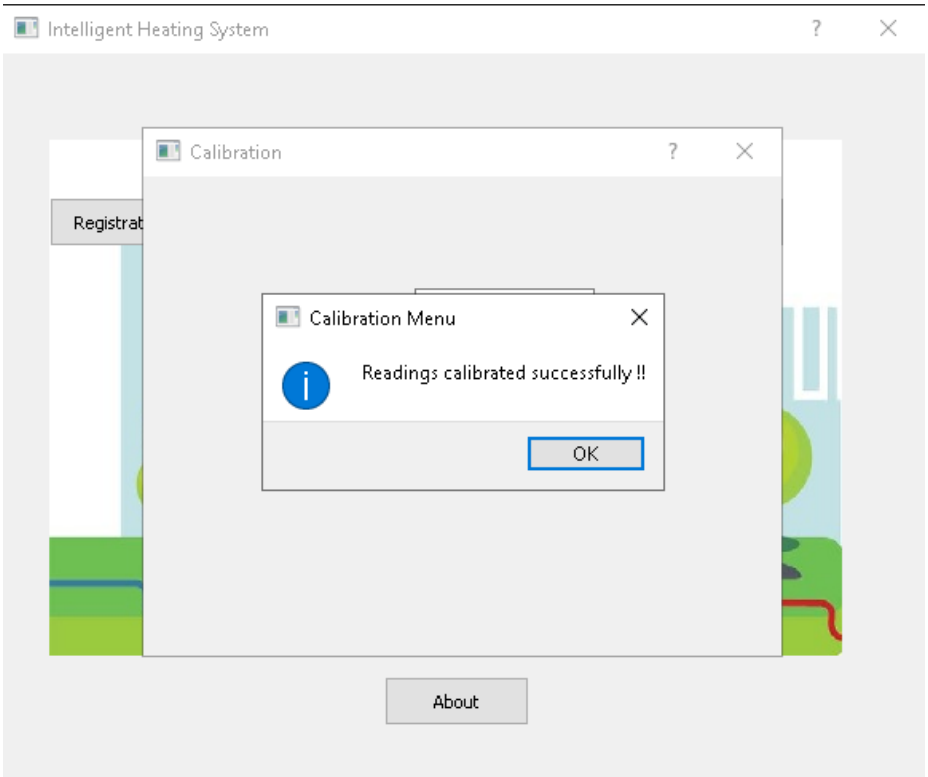
After multiple calibration system will predict next value based on user's wish.



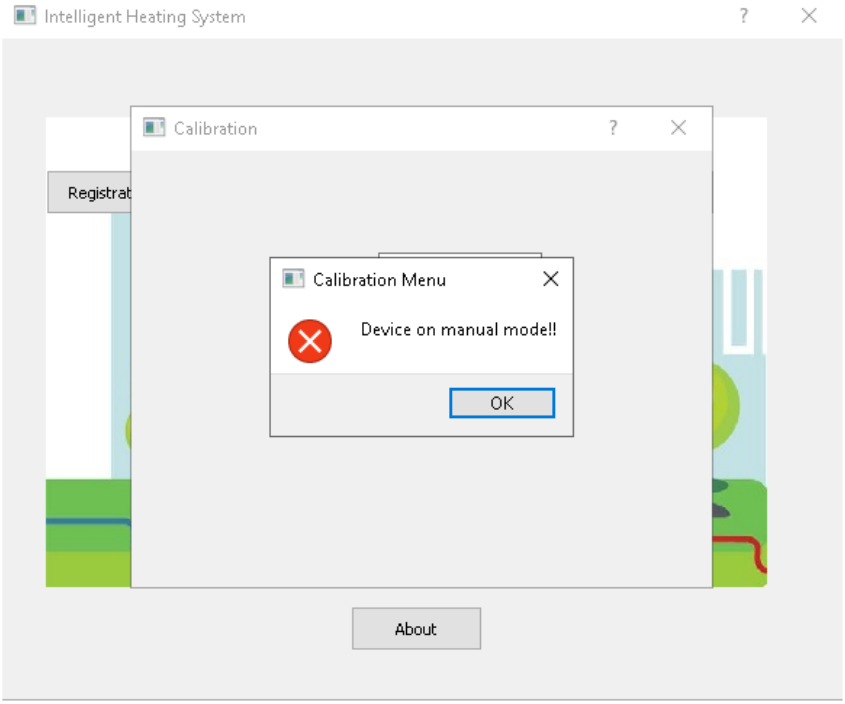
If yes then system will use machine learning algorithm to predict:



If no then it will drive back the system on calibration mode



However, If three predictions by system is not liked by user then system will turn to manual mode forever

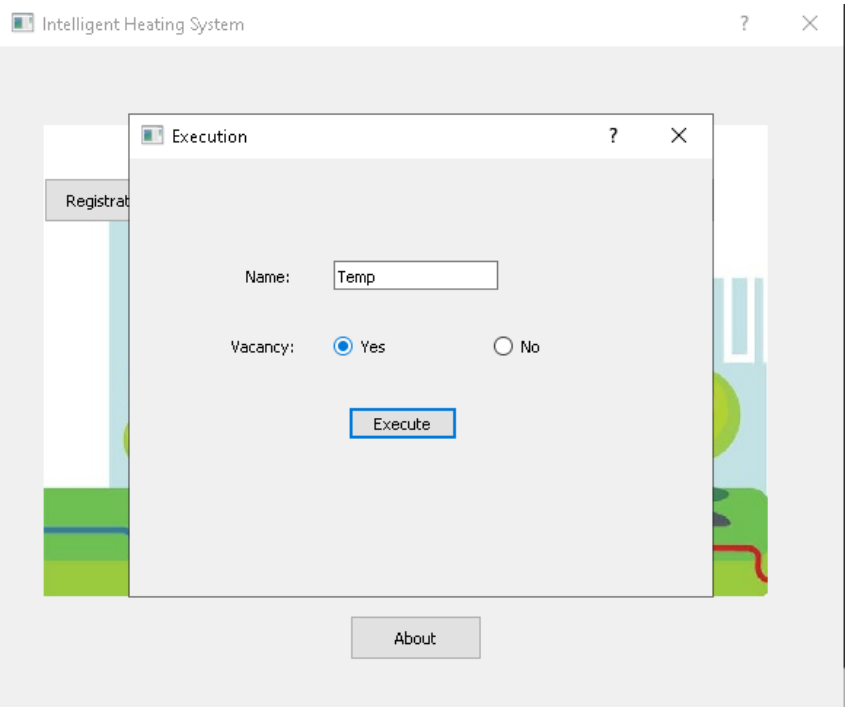


In this mode user can still calibrate, but system will never use machine learning algorithm

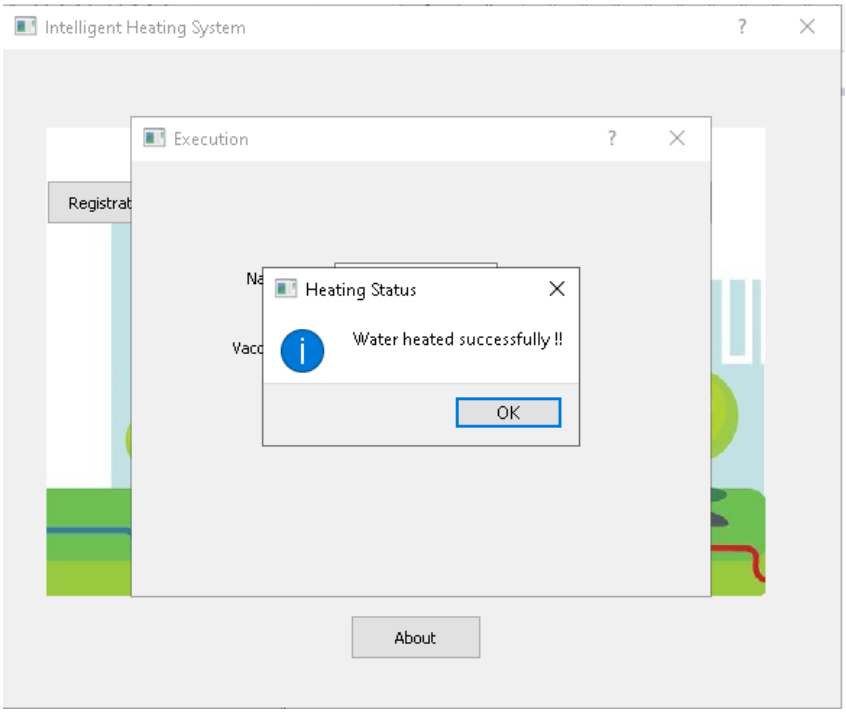
Execution:

Will finally give command to heater to operate. It will first ask from user to ensure bathroom is vacant.

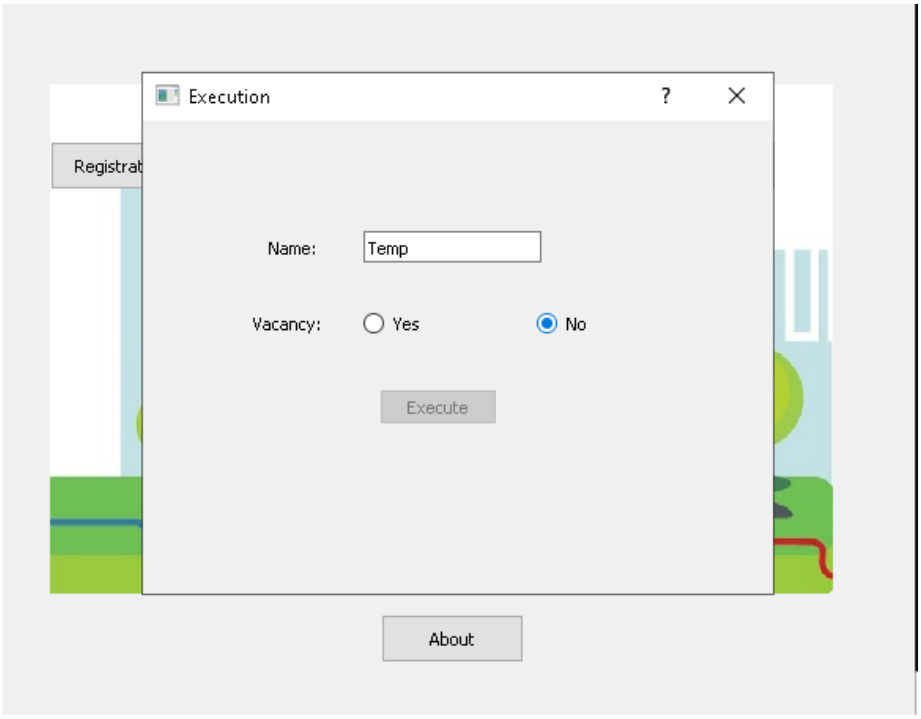
If yes then execute button will be enabled:



When execute is clicked, then system will operate heater, till the time process is happening execute button will be disabled, so that anyone else can't give command. After completion:



If bathroom is not vacant then execute is disabled:



About:

This button will be linked to the website which will provide necessary help if user needs.

Software Back End

We will visualize system as per real world scenario.

Devices (Robot) in the system:

So, the database is like shelf on which records are kept.

The shelf-Operator robot keeps and takes the records from this shelf.

The iot-Operator robot operates the Wi-Fi module(IoT device), it calibrates and operate heater up-to temperature required

The machlearn robot is intelligent one! It implements prediction algorithm, if it fails then guide system to manual mode.

The shelf(database) looks like:

name	col1	col2	col3	ctr	latest	col4	col5	col6
Tantu	24	NULL	NULL	1	24	NULL	NULL	NULL

Windows of Intelligent Heating System:

To visualize system, we could imagine all the windows to be humans! So, the first window is like reception, it directs user to respective counters, i.e. windows, like registration, deletion etc. Let's first go to registration counter.

Registration:

Now the person sitting here asks for user's name. Then counter man asks iot-Operator to calibrate user's reading and then asks shelf-Operator to put user's record in the shelf(database).

Execution:

Now in execution counter user goes. Counter man asks his name and then he proceeds. He asks shelf-Operator to give user's "latest" calibration and then with that value asks iot-Operator to switch on heater.

Calibration:

Now the user went to bathroom but was not happy with the water temperature. He decides to recalibrate. Then counter man asks iot-Operator to calibrate user's reading and then asks shelf-Operator to put user's record in the shelf(database). Shelf-Operator puts it in col2 and latest holds value of col2.

If it again continues then after col3 system will offer user for prediction. If user agrees then machlearn robot will predict and update latest column, even then if user is unhappy then he can again calibrate. But this time shelf-Operator will place in col4.

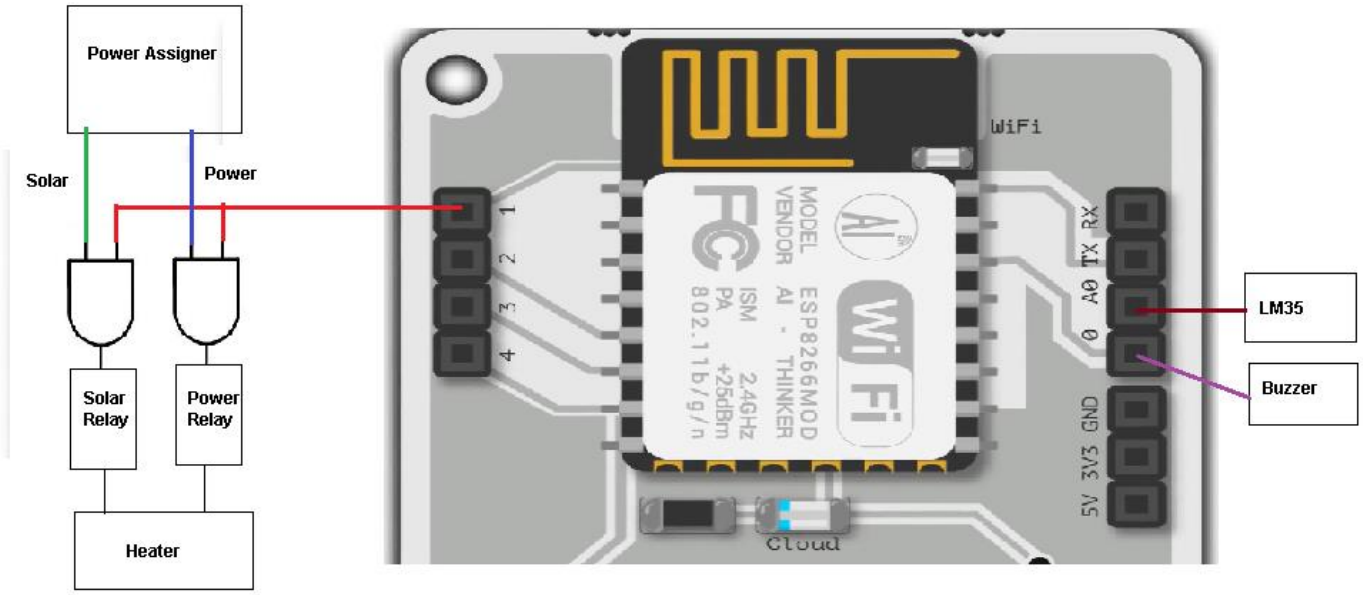
The system will offer user prediction for 2 more unsuccessful calibration. After that system will be on manual mode and user can although calibrate but now prediction algorithm will not work.

Deletion:

Suppose a guest came and now he/ she has gone. Then user can go to deletion counter. Their counter man will ask shelf-Operator to remove the user record from shelf(database).

For software code refer Appendix

Interfacing with Wi-Fi Module



Appendix

Following pages contain software codes of all the modules in the project