Faculty of Engineering and Applied Science

SOFE 4610U Design & Analysis of IoT Software

Assignment 2

Group #3

Group Member 1

Name: Sarthak Sharma

Student ID: 100604428

Group Member 2

Name: Adam Wong Chew Onn

Student ID: 100598499

Group Member 3

Name: Mitul Patel

Student ID: 100700131

Group Member 4

Name: Matthew English

Student ID: 100704553

10 Step Approach

1.Purpose & Requirement Specification:

Purpose: The purpose of our project is to create a smart cooling system.

Behaviour: The system will be able to sense the temperature and decide whether the fan is turned on or off.

System Management Requirement: The system should provide local monitoring and control functions.

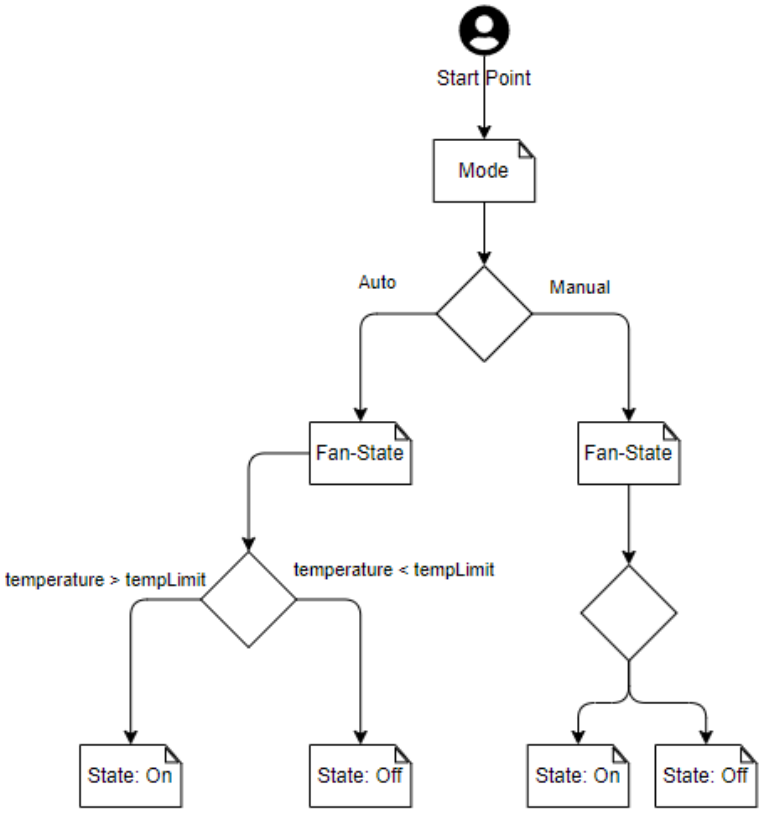
Data Analysis Requirement: The system must read and display the active temperature to the user

Application Deployment Requirement: The application should be deployed locally with a

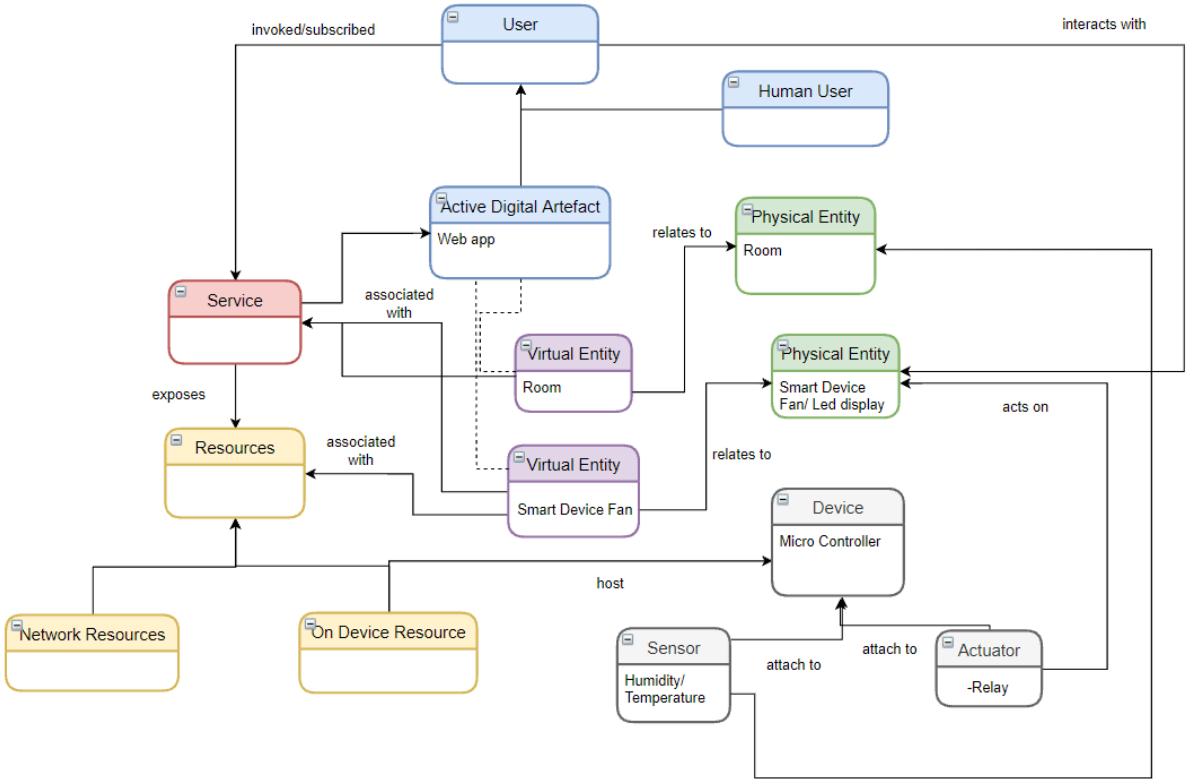
User Interface Requirement: The user should be able to see the current temperature and manually turn the fan on and off

Backend Requirement: The data should be set from the microcontroller to the django backend using a post request. Furthermore, the application should display temperature data on the app.

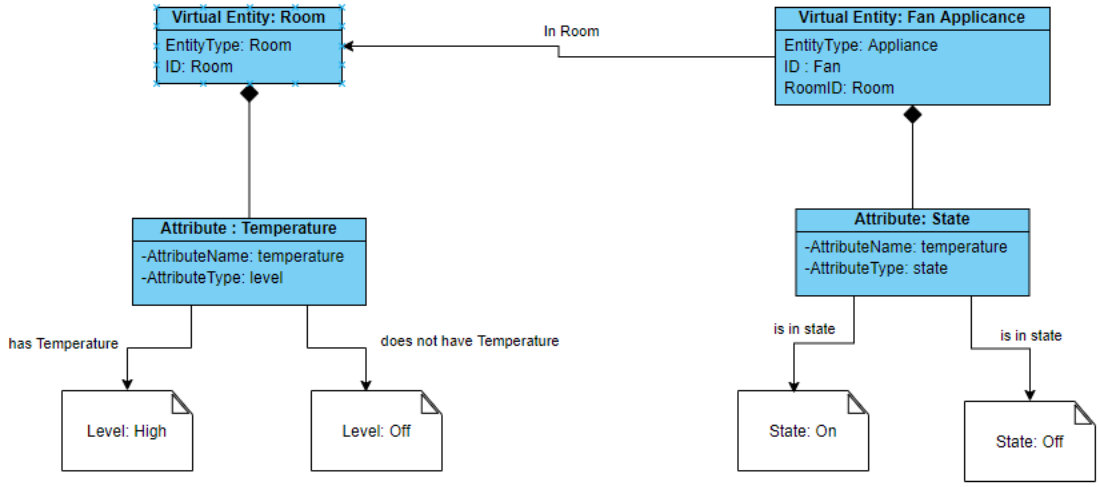
2.Process Specification:



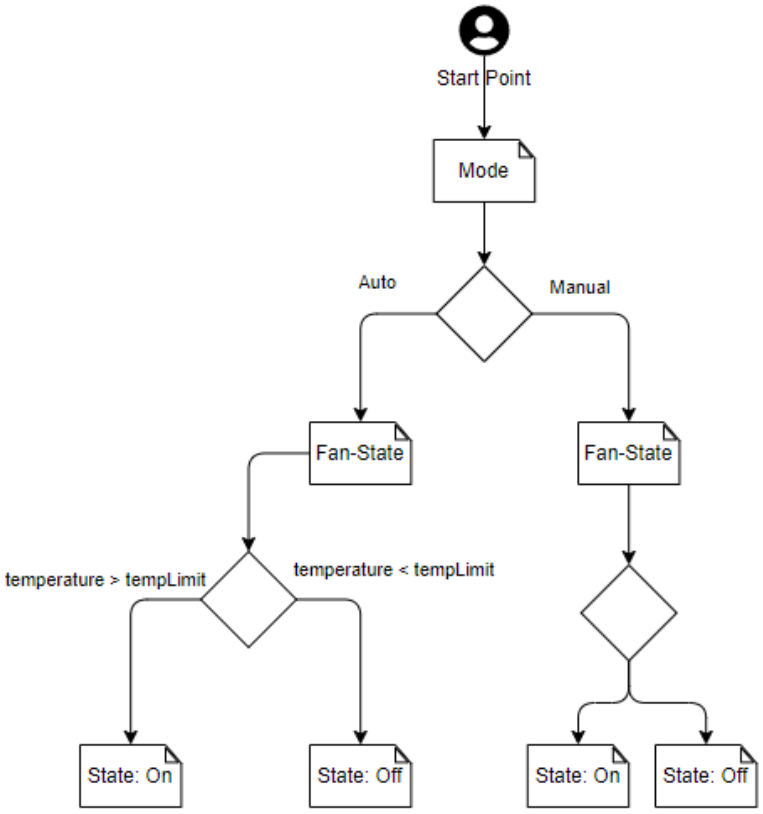
3.Domain Model Specification:



4.Information Model Specification

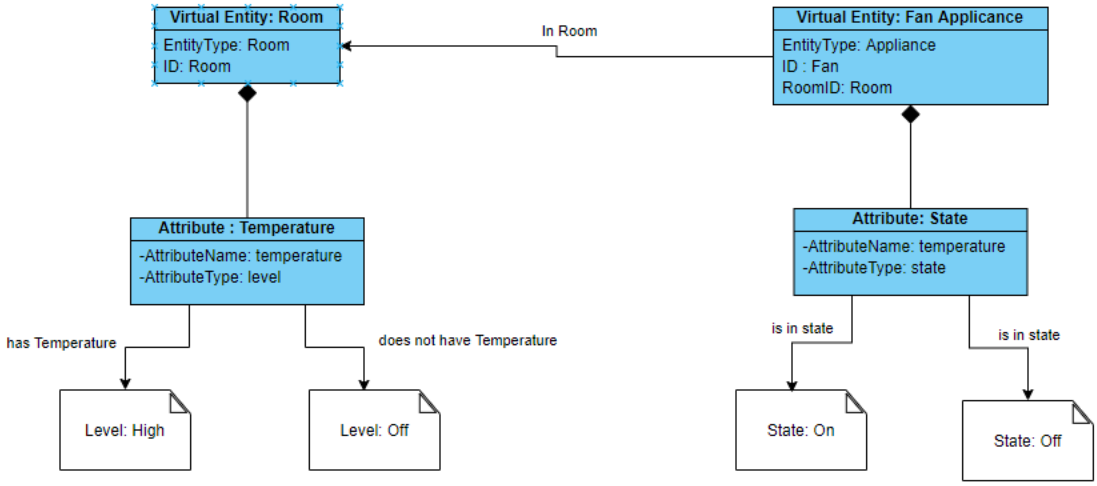


5.Service Specifications:



Mode Service allows the user to manually turn the fan on and off instead of the auto mode.

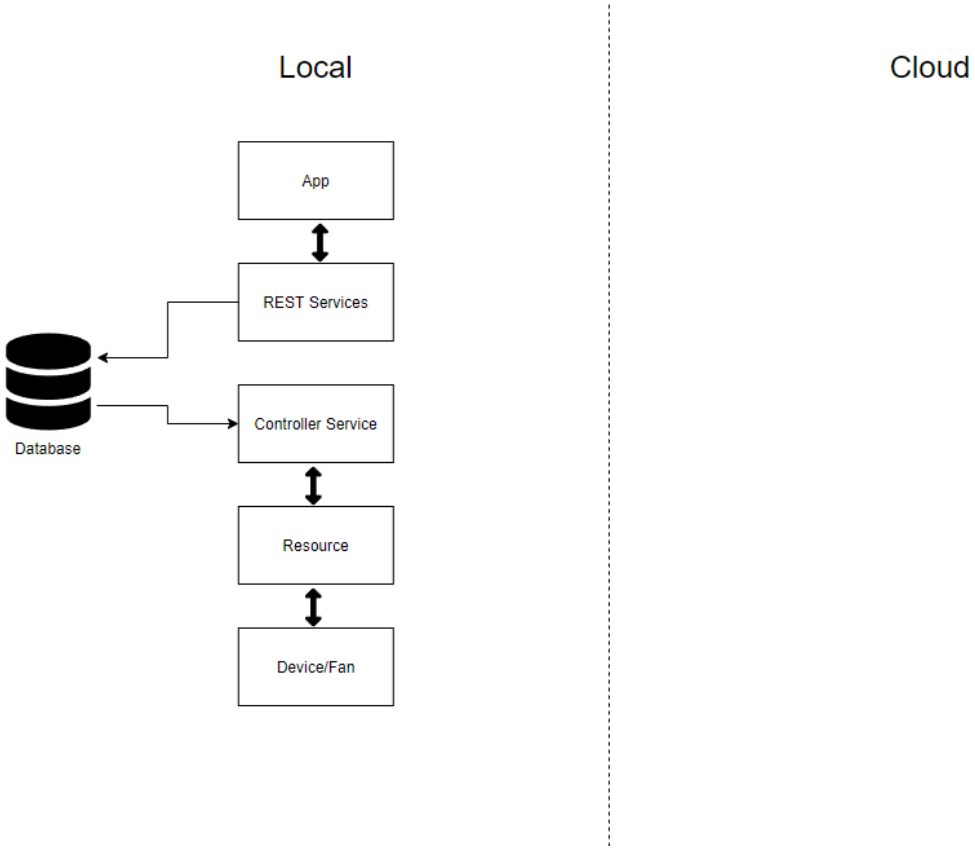
State Service: Sets the fan appliance on and off depending on current temperature.



State Service: Sets the fan appliance on and off depending on current temperature.

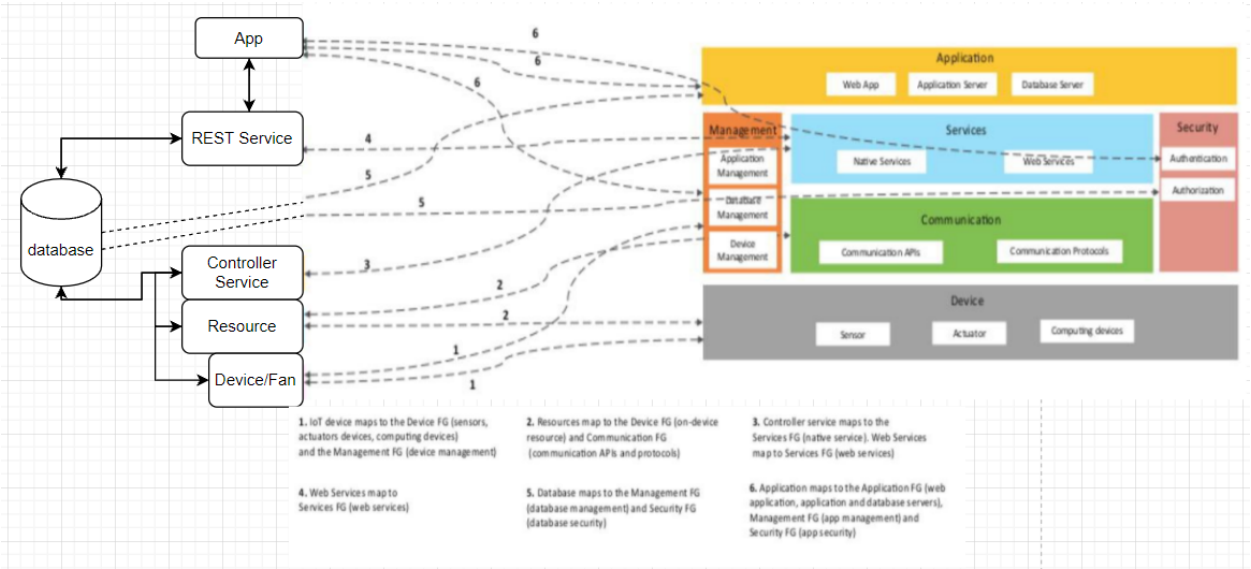
Controller Service: Auto mode monitors the temperature and switches the fan on and off. In manual mode, the user is able to change the state of the fan by switching it on/off.

6.Iot Level Specification:



7.Functional View Specification:

Diagram



User case:

UC 1- User should be able to view their device information from the application

UC 2- Devices must communicate with Django applications with high level api rest calls.

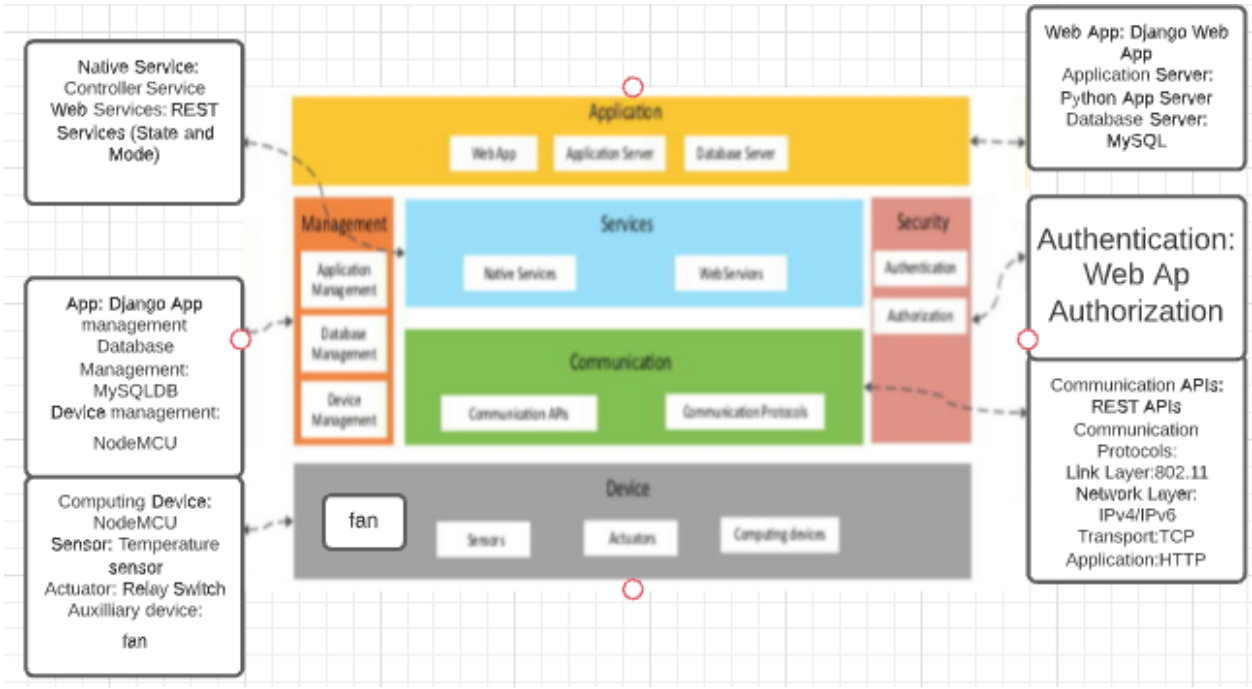
UC 3- Users should be authenticated using tokens and app should manage users

UC 4- The device should turn on the fan when the temperature passes a certain range.

These use cases are mapped to the application developed below. The mappings are available on the github along with the project: **https://github.com/Mitul2000/Smart_Cooling_System_IoT**

8.Operational View Specification:

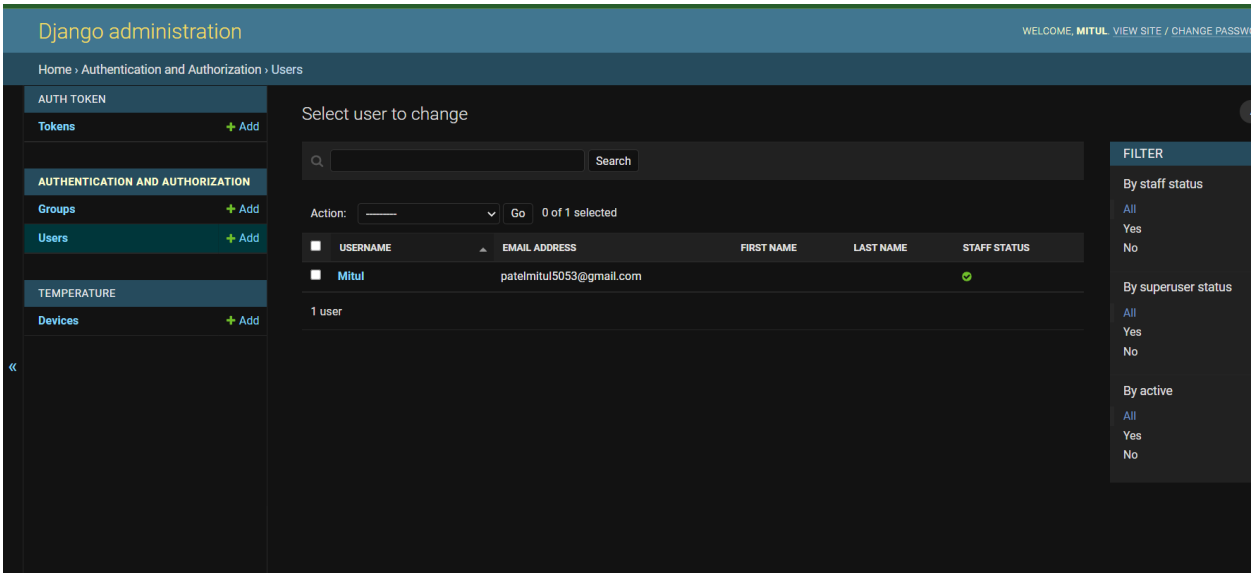
Diagram



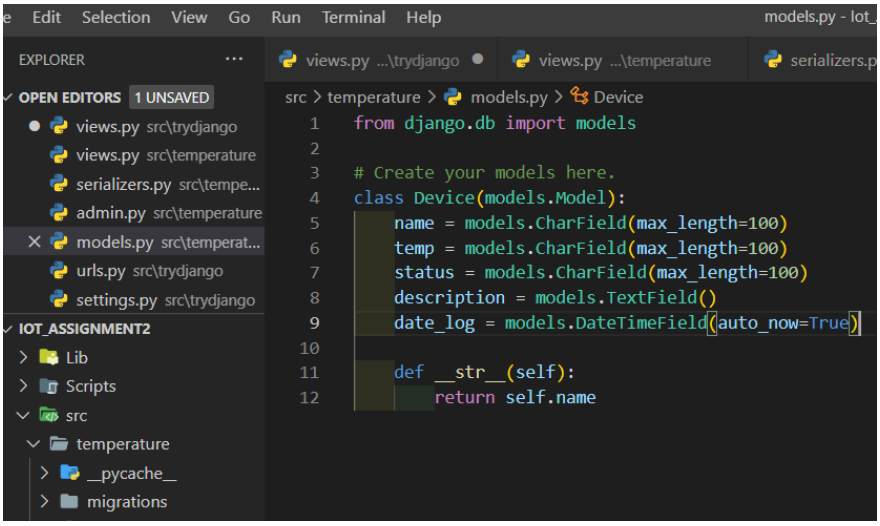
2.Implement your design using the Django REST framework. Submit a GitHub link of the code and tests for the implementation. In the README file create a table of contents to help navigate your code and show that your implementation performs the functionality if the use case is picked for design.

Github link : https://github.com/Mitul2000/Smart_Cooling_System_IoT

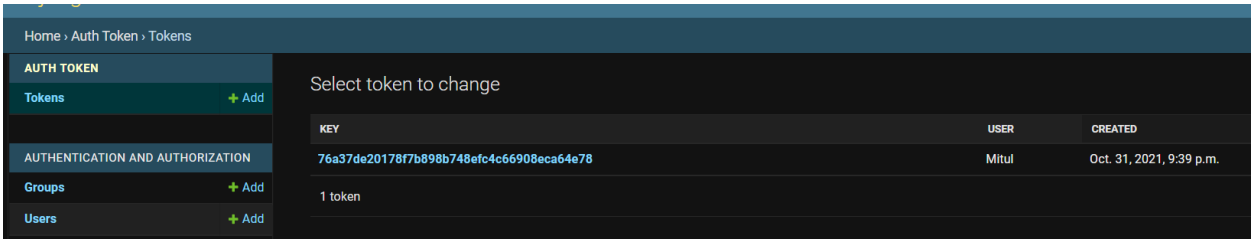
Admin page:



Device Model

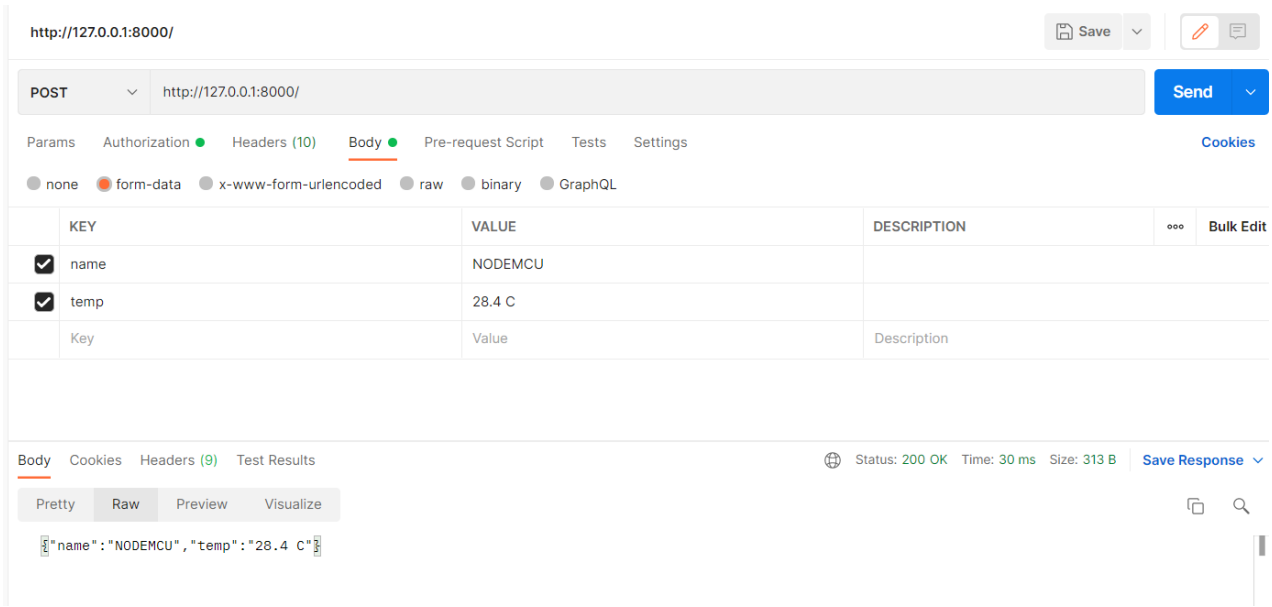


User Token:

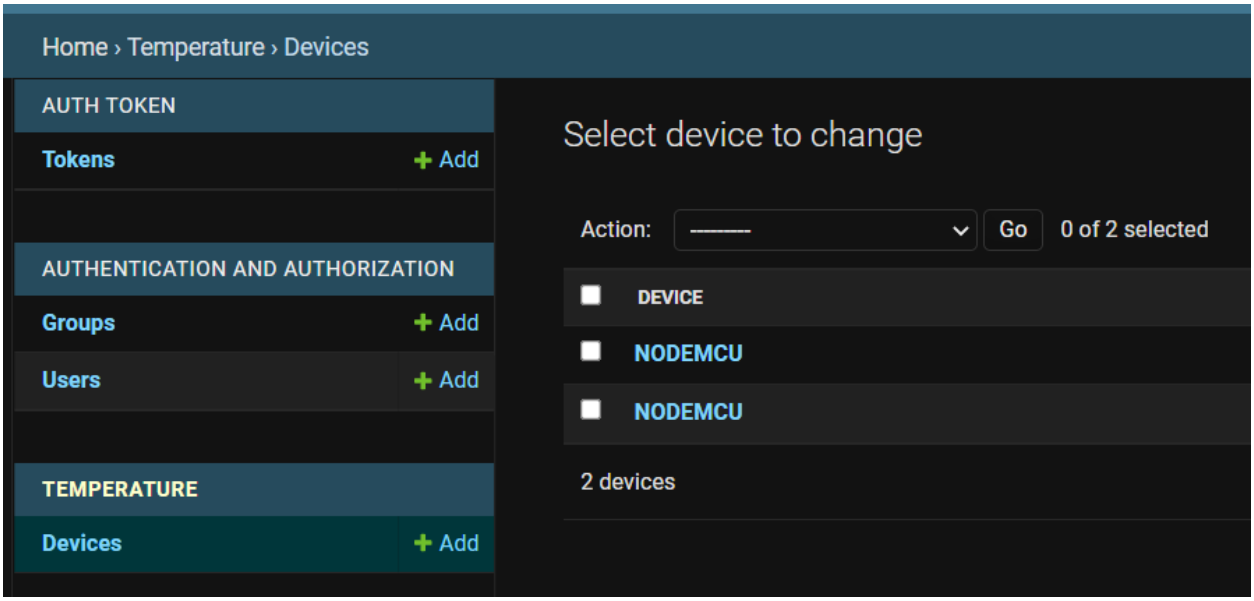


Test 1:

Making POST Request with authentication token:

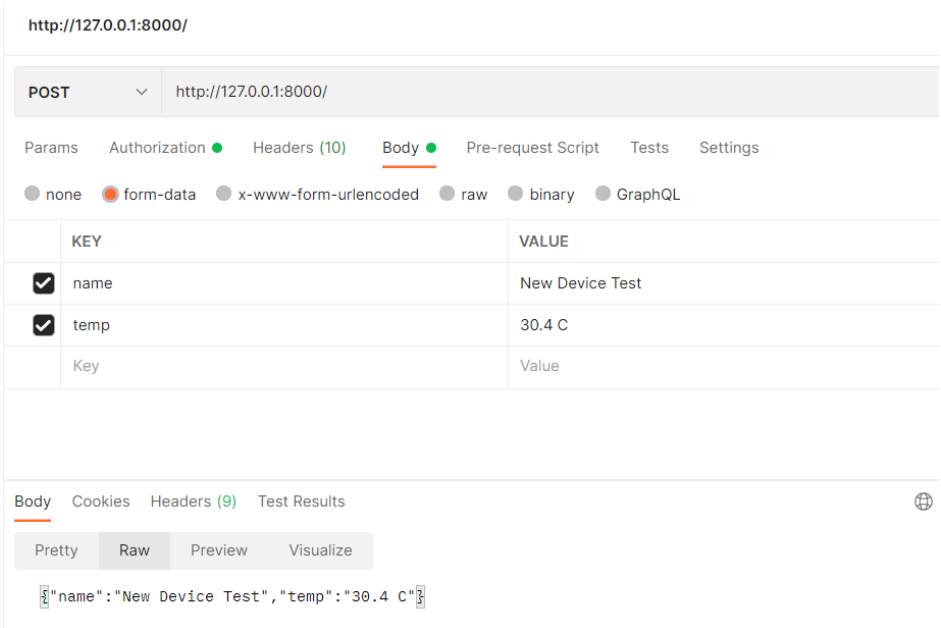


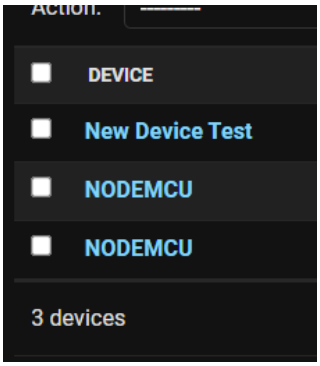
Device Object in admin view



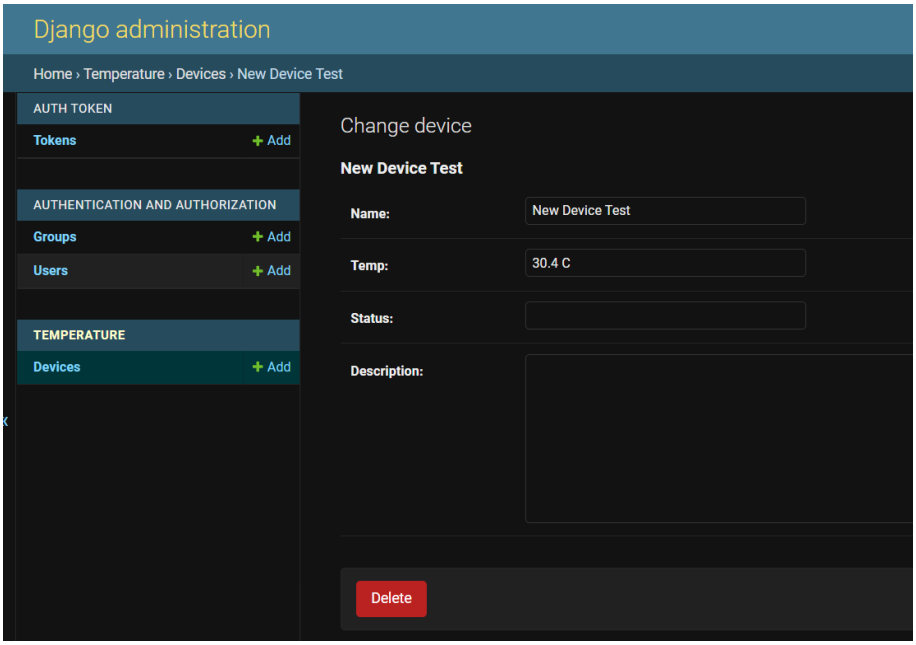
Test2:

Making POST Request with authentication token for registering new device:

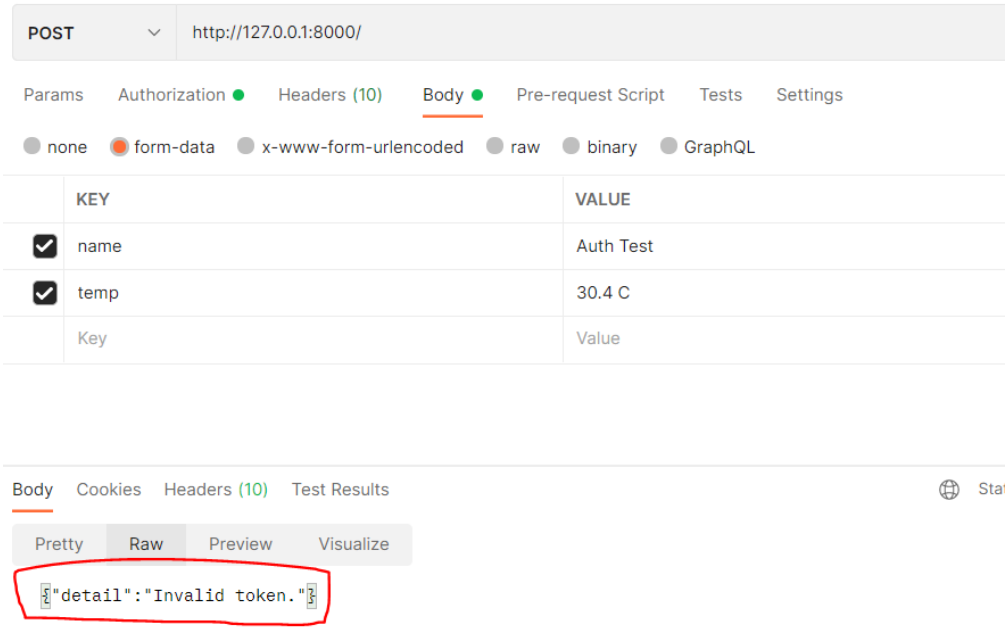




New Device attributes:



Test 3: sending post request with authentication token



Test 4:

Get request to view all devices and temperature on dashboard page:

This list can also be filtered based on the user.