Faculty of Engineering and Applied Science

SOFE 4610U Design & Analysis of IoT Software

Final Project Proposal

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

IOT Smart Cooling System

Short project description:

Our IoT project will be to create a smart Cooling system. The idea is to design and create a smart fan which will sense the temperature at every certain interval to assess if the fan needs to be turned on or not. The Fan is connected to a DC motor and will be connected to the NODE MCU microcontroller. Furthermore, we will also implement an alphanumeric display which will tell the user what the current temperature is. The backend will allow all multiple devices to connect to the smart fan. Using the pub/sub architecture, we will pass sensor data from the smart cooling system to a broker and from the broker, we use a Django REST API to store and collect data from the controller to other devices. Most of the logic will occur in the far edge of the system and specifically for our case, it will happen on the microcontroller itself.

Hardware Requirements:

- **DHT11 Temperature & Humidity sensor module, DC motor, Battery holder box, I2C LCD Display (16x2), Solderless breadboard, Infrared Remote Controller, Infrared receiver.**

Functional Requirements:

Requirements Gathering

REQ-ID/ CON-ID	Type	Requirement Specification	Priority
REQ-1	Functional	System should sense the temperature at every 10 seconds interval	HIGH
REQ-2	Functional	System should calculate if the temperature received is above user's preference and turn on the fan accordingly	HIGH
REQ-3	Functional	System should display the temperature on Alphanumeric I2C LiquidCrystal Display(16×2 LCD)	HIGH
REQ-4	Functional	System should be portable with replaceable batteries unit	MEDIUM
REQ-5	Functional	System could be manually controlled using Infrared remote controller	LOW
REQ-6	Functional	System should have a power off Switch	LOW
REQ-7	Functional	Users should be able to control the	MEDIUM

		system using a webclient.	
REQ-8	Function	The fan should have a 10 timer where it should not be interpreted after the sensor signals to turn on.	MEDIUM
REQ-9	Non-Functional	The system should turn the fan after detecting the temperature within 5 seconds of response time	MEDIUM
CON-1	Non-Function	System should establish connection to a broker in under 2 seconds	HIGH

Backend Requirements

REQ-ID/ CON-ID	Type	Requirement Specification	Priority
REQ-1	Functional	Implement a communication protocol to connect the device to a broker. (MQTT),(COAP)	HIGH
REQ-2	Functional	Allow multiple devices to communicate with the fan for better control	MEDIUM
REQ-2	Non-Functional	Implement a SSL/TLS certificate for secure connections	LOW
REQ-3	Functional	Implementing Backend using Django using REST APIs	MEDIUM
REQ-4	Functional	Store data into a database such as Sqllite	HIGH
REQ-5	Non-Functional	The website should have a high availability	MEDIUM
REQ-6	Non-Functional	The website should only allow authenticated users with token to access device models	HIGH
REQ-7	Non-Functional	Each client should only receive data associated to their device	HIGH
REQ-8	Functional	Admin should have the ability to see all the devices available	MEDIUM