Faculty of Engineering & Applied Science

SOFE 4610U Design And Analysis of IoT Software Systems

Smart Kitchen Ventilation System

Architectural Design Report

Github: https://github.com/Waleed20210/IOT-Project

Deadline date: 12/02/2022

Group Number*: 4*

Course Instructor: *Ramiro Liscano*

Student Name	**Student Id**
Preet Patel	*100708239*
Tiwaloluwa Ojo	*100700622*
Waleed El Alawi	*100764573*

Introduction

Ventilation System is extremely important to regulate the airflow in the kitchen. A ventilation system will circulate the air and remove the smoke, smells, steam, and particles which can irritate eyes and cause respiratory problems. The goal of our project is to create a smart kitchen ventilation system to modernize the existing manual ventilation systems. For this project, we will add a temperature and humidity sensor to a ventilation system (fan) which can operate in either manual or automatic mode. The manual mode will allow the ventilation system to behave like the existing ventilation systems in the market which turn on and off from a button click. In our case, the user will use the system web dashboard to manually control the fan. In the auto mode, the ventilation system will use the temperature and humidity sensor to read the room temperature and humidity and if the reading values rise above the set safety threshold then the fans will turn on automatically. If the reading value falls below threshold then the system will turn off the fans automatically.

Purpose and Requirements

– Purpose : The purpose of this project is to create a smart kitchen ventilation system which assists the homeowner to turn on/off the ventilation fan if the temperature or humidity of the room increases.

– Behavior : The system should be able to detect increases in temperature and humidity in the kitchen and should turn on the ventilation fan.

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

– Data Analysis Requirement : The system should sense room temperature and humidity and if any increases in temperature and humidity are detected then it should notify the user and turn on the ventilation fan.

– Application Deployment Requirement: The application should be deployed locally.

– Application Requirement : The system should turn on the ventilation fan and notify the user if any increases or decreases in the temperature or humidity are detected by the system.

Requirements

Stakeholders	Requirement ID	Requirement Type	Requirement Description	Priority
Resident/Landlords	REQ 1	Functional	System should detect the temperature and humidity of the room.	H
Resident/Landlords	REQ 2	Functional	System should be able to turn on the fan automatically when the temperature is higher than 30 degrees.	H
Resident/Landlords	REQ 3	Functional	System should notify the user if increases in temperature and humidity are detected.	H
Resident/Landlords	REQ 4	Functional	Users should be able to control the state of the fan remotely.	H
Resident/Landlords	REQ 5	Non-Functional	System should be able to work with both wifi and bluetooth.	M
Resident/Landlords	REQ 7	Non-Functional	The system will utilize MQTT protocol for stable low bandwidth event streams	H

Use Cases

UC-1: Temperature and humidity detection

1.1. **Description:** The temperature and humidity sensor detects increases in temperature or humidity in the kitchen.

1.2. **Actors:** Property Owner/Resident

1.3. **Basic Flow:**

1. Temperature and humidity sensor on the system detects an increase in temperature and humidity of the kitchen.
2. System notifies the user of the increase in temperature and humidity and prompts the user to turn on the fan.
3. System automatically turns on the fan if temperature or humidity rise above a threshold level or the user turns on the ventilation fan in the system remotely using the application.

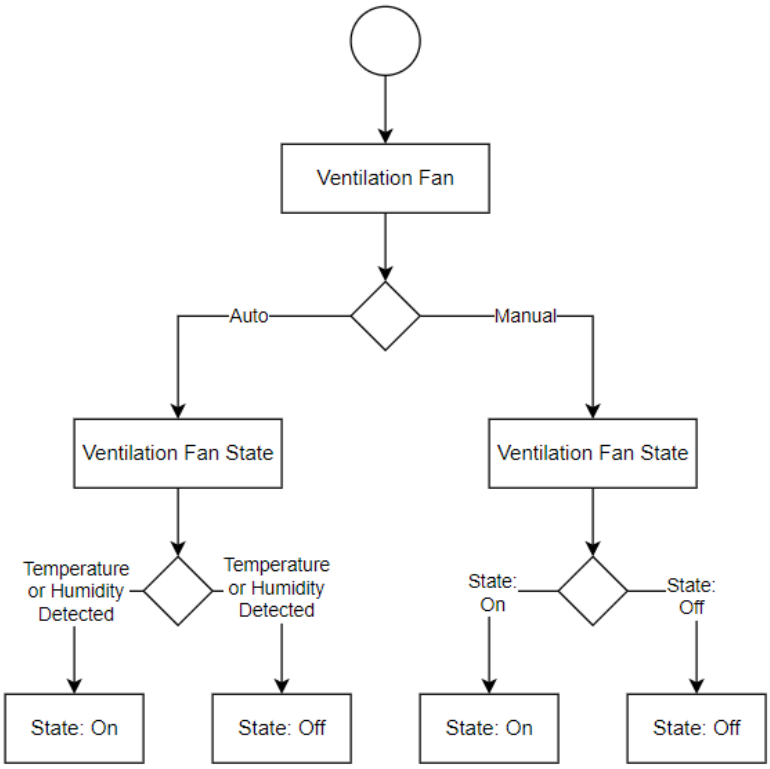
1.4. **Post Condition:**

1.4.1. **Successful Condition:** Sensor detects temperature and humidity, user receives notification of the event from the system, the fan is turned on manually (remote) by the user or automatically by the system

1.4.2. **Failure Condition:** System is unable to detect temperature or system, system is unable to send event notification to the user, and system is unable to turn on the ventilation fan manually or automatically

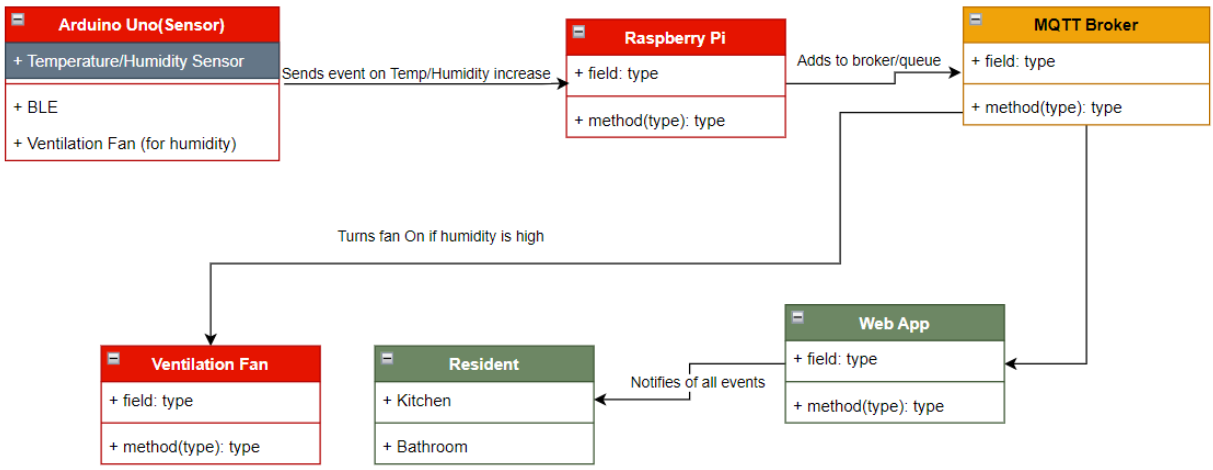
Process Specification Model

The process specification model is used to capture the use case. This model captures the temperature and humidity which will be used to turn on the ventilation automatically if the temperature is higher than 30 degrees or manually by the user.



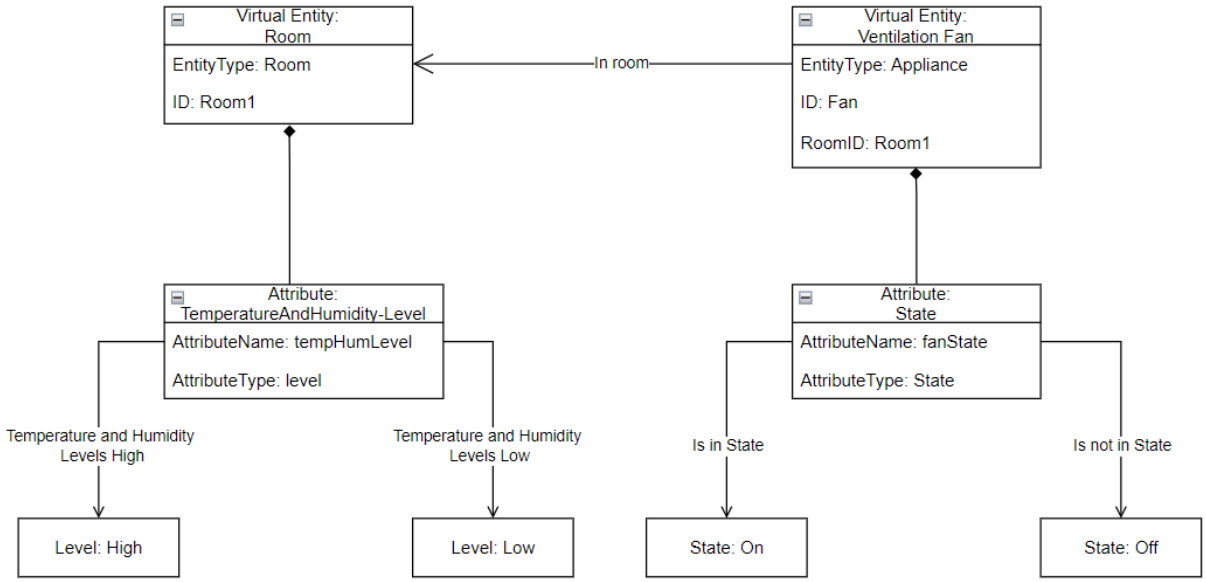
Domain Model Specification

The main concepts here are to design and implement a smart kitchen ventilation system that detects temperature and humidity which will assist the homeowner to be notified in case the temperature changes. The main objects for this project are sensors that would be used to detect the temperature and the humidity.



Information Model Specification

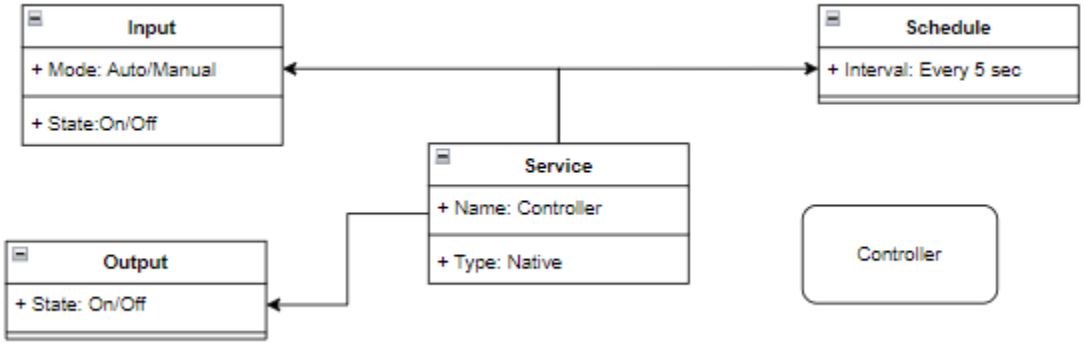
This model describes the structure of all the information in the systems. It looks at the temperature and humidity attributes inside of the room entity and the state attributes of a ventilation fan.



Service Specifications

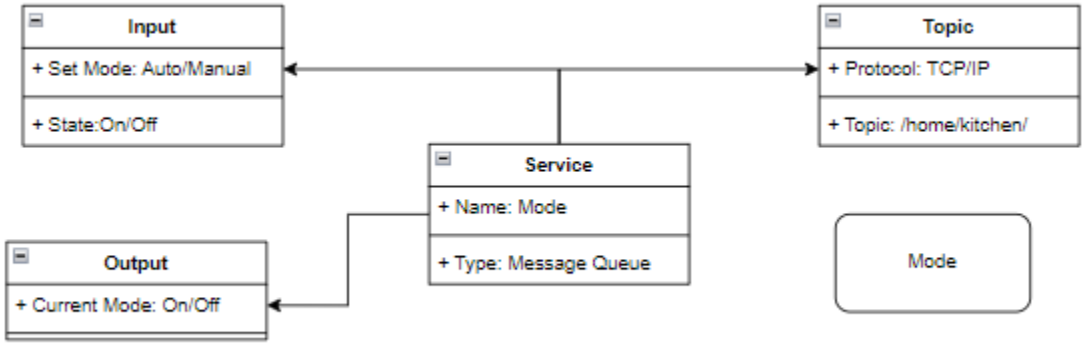
Controller Service:

In auto mode, the controller service will monitor the temperature and humidity of the room and automatically turn on the ventilation fan if the temperature or humidity increases above the threshold level. In manual mode, the service retrieves the fan state from the user application to switch the fan on/off.



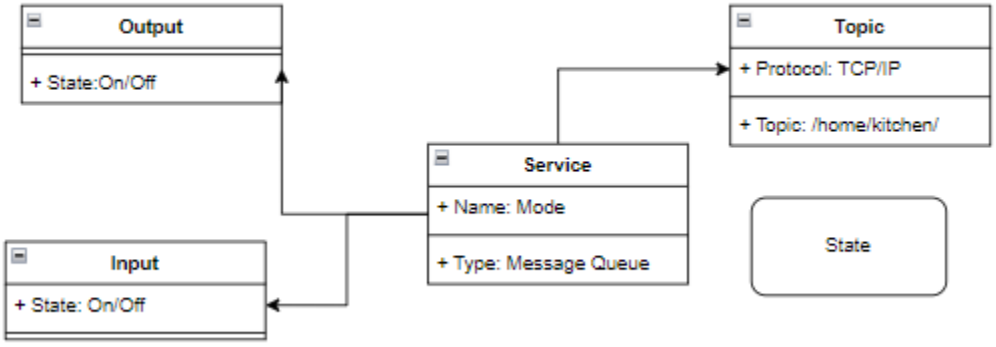
Mode Service:

The mode service will retrieve and set the current mode of the system to auto or manual.



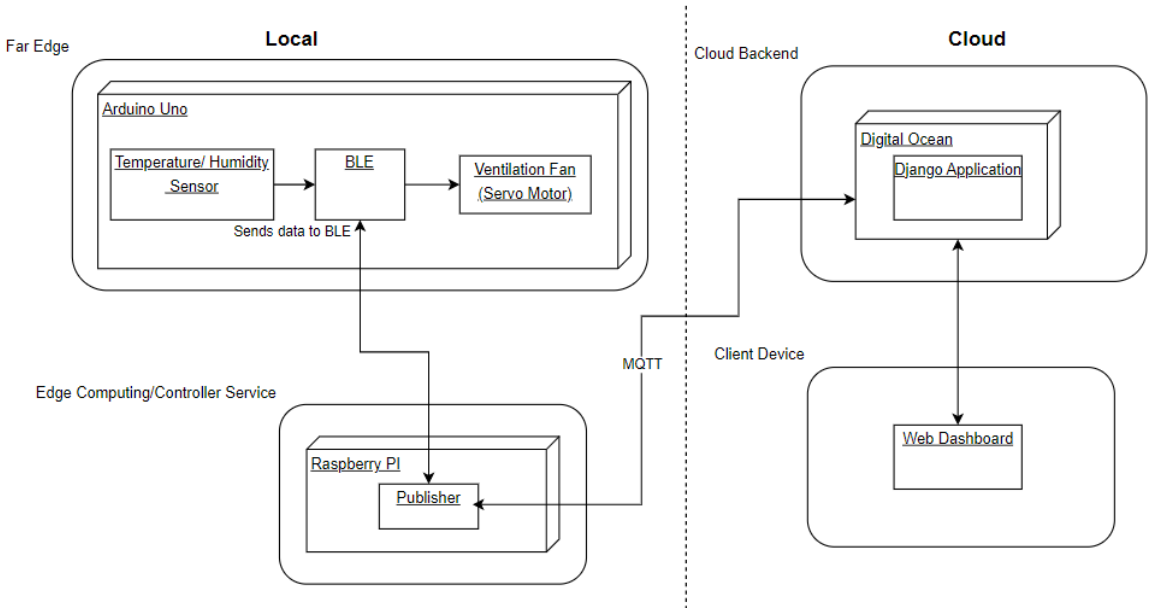
State Service:

The state service retrieves the current state of the fan and sets the state of the fan to off or on.



Deployment Diagram

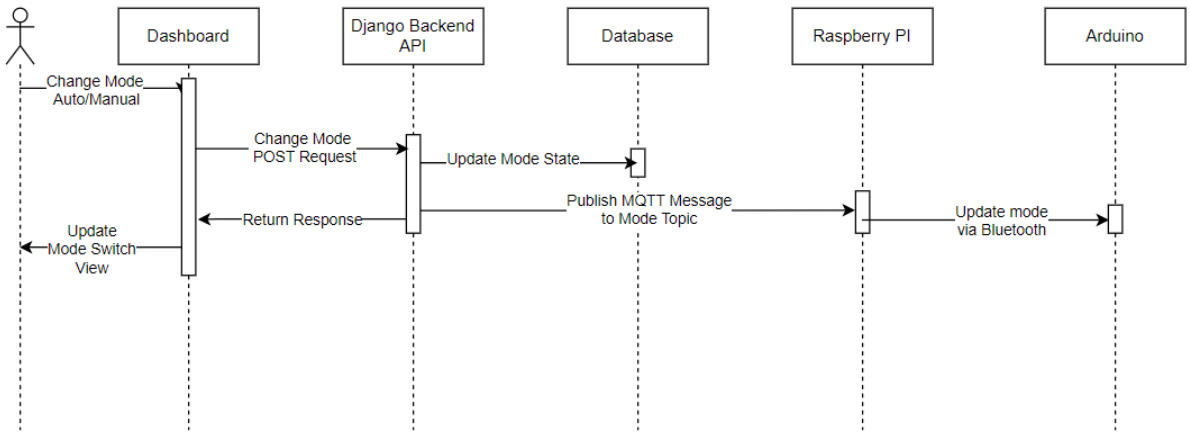
The deployment diagram explains the deployment of different components in the system and how these components will interact with each other.



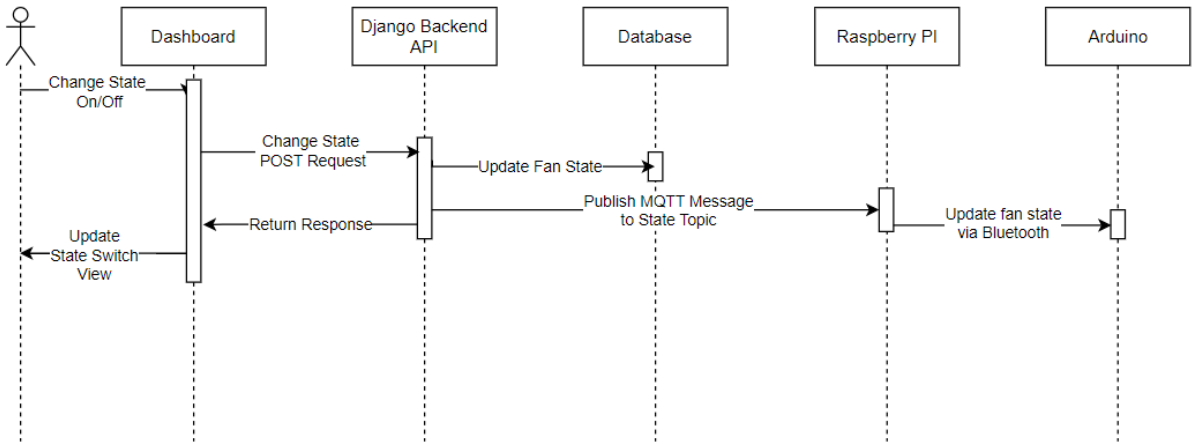
Sequence Diagrams

The following sequence diagrams describe the different processes in the smart ventilation system:

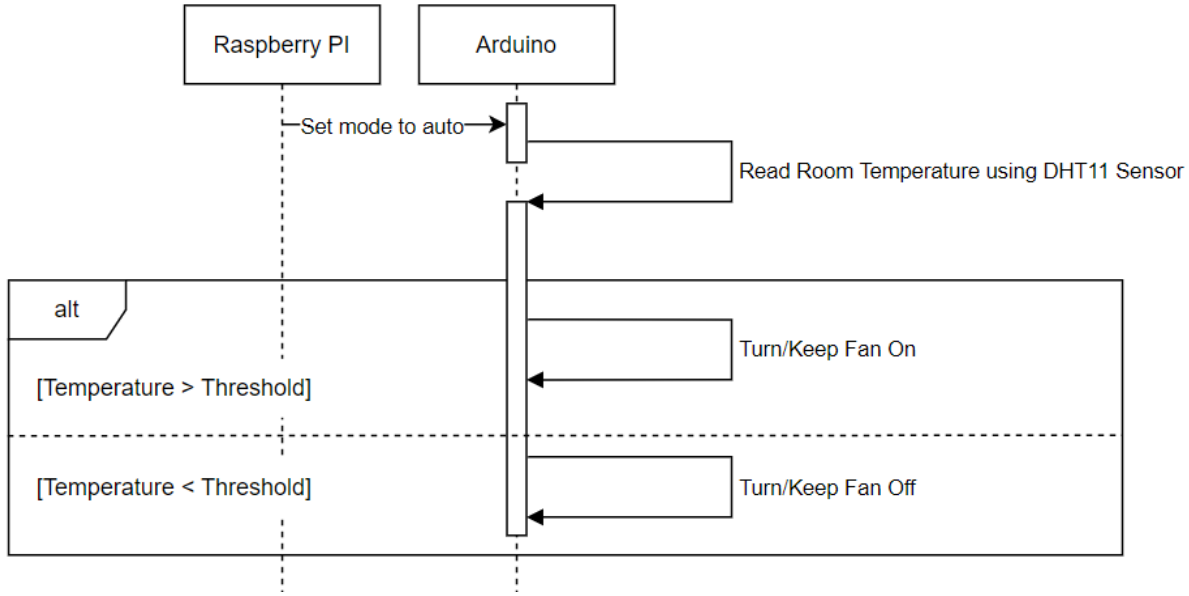
1. Switch Mode Sequence Diagram



The sequence diagram above describes how the system will update the mode to either manual or automatic. To start the process, the user can switch between the manual or automatic mode from the web dashboard which will create a POST request to the backend Django application API. This API will first create an update query to change the current value of the mode in the database and then it will publish a MQTT Message on the Mode Topic. The Raspberry Pi will be subscribed to this Mode Topic allowing it to receive the new message which it will then forward to the Aruduino to change the mode of the ventilation system device (fan). After this the API will return a message to the web dashboard, which will then update the view of the mode switch to auto/manual.



The sequence diagram above describes how the system will update the state of the fan to either on or off when the system is in **manual mode**. The user will control the state of the fan from the web dashboard. When the user changes the state, a POST request will be made to the state API in the Django application, which will first update the state value in the database and then publish a message on the State Topic. The Raspberry PI will be subscribed to this topic and will forward this message to the Arduino using bluetooth to change the state of the fan to on/off. Then the API will return the response status and message to the dashboard application which will in turn update the state switch to on/off.



The sequence diagram shows how the system will function when the system is set to auto mode. In this mode, the Arduino will constantly poll the temperature and humidity (DHT11) sensor for the temperature and humidity reading of the room. If the temperature or humidity is higher than the set threshold then the system will automatically turn on the fan or keep the fan on if it is already running. However, if the temperature or humidity is below the threshold then the system will turn off the fan automatically or keep the fan off if the fan is not running currently.