Deakin University

Software Architecture And Scalability For Internet-Of-Things

OnTrack Submission

# Project Presentation

*Submitted By:*
Deol Satish
dsatish
2021/10/11 20:59

*Tutor:*
Purnima Das

| **Outcome** | **Weight** |
|---|---|
| Demonstrate | ♦◊◊◊◊ |
| Apply | ♦◊◊◊◊ |
| Develop | ♦◊◊◊◊ |
| Deploy | ♦◊◊◊◊ |

Task is good

October 11, 2021

# Smart Lighting System

SIT 314- Project Presentation

# Problem Description

- Usually lighting control has been manual, which might require physical, local control manipulation and is commonly used in the widest scope of applications. Even these days many buildings and houses do not have smart lighting control and have do switched on and off physically and easy to forget that the light is On and there might be huge wastage of electricity. This happens everywhere and all the time.
- One of the main players in global energy consumption are buildings, they make up to 30% of the global energy consumed on the planet and 60% of the electricity produced.
- 20% of the total consumption of energy produced and 30% of the electricity consumed in buildings is because of lighting related to residential and working environments. Any energy consumption in building illumination is associated with the type pf lamp, its efficiency , and mode of use, among others.
- “Optimizing the energy consumption of lighting systems depends not only on how efficient the technologies with which the lamps and the electrical elements that make up it are developed, but also on the techniques associated with the controllers of these lamps”.[1]
- Nowadays, we have better smart lighting systems, they are gaining popularity but not as fast as we want it to be. One of the main issues is that it is not easy to use and are made as consumer products meant for houses and they are not cheap, not easily compatible with other light devices from different manufacturers. In one report it mentioned that because of the smart system being online 24x7, security becomes a concern.

Thus, smart lighting has evolved to integrated control systems that allow the designer to connect sensors, controllers and user interfaces for the development of applications such as intelligent RGB lighting , habitability scenes, advertising and promotion, light decoration, shows, etc. Also additional features for intelligent lighting control systems currently include energy management, reporting of breakdowns, energy consumption reports, positioning, time programming and scheduling, voice control, local and remote control, etc. [1]

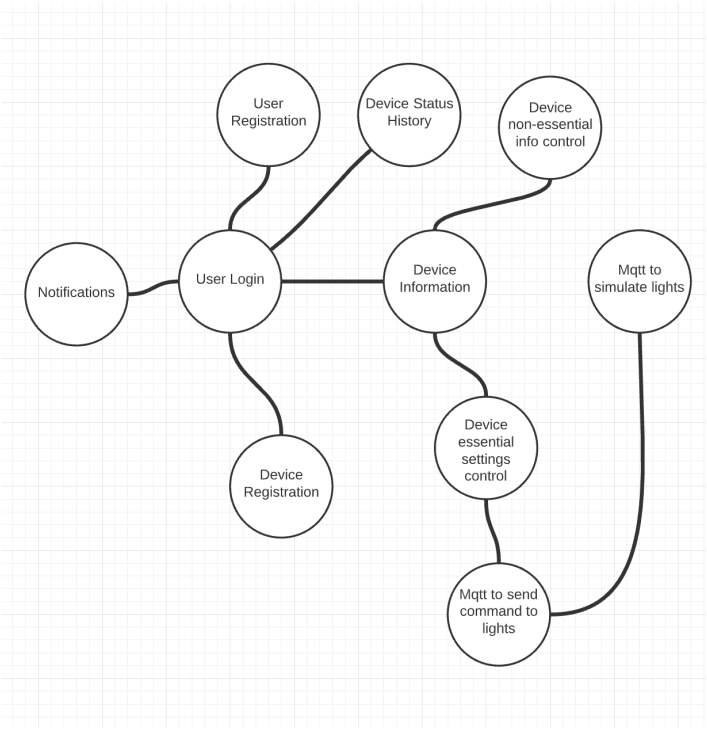

Fig. 1. Smart Lighting in intelligent environments.

# Some methods we can use

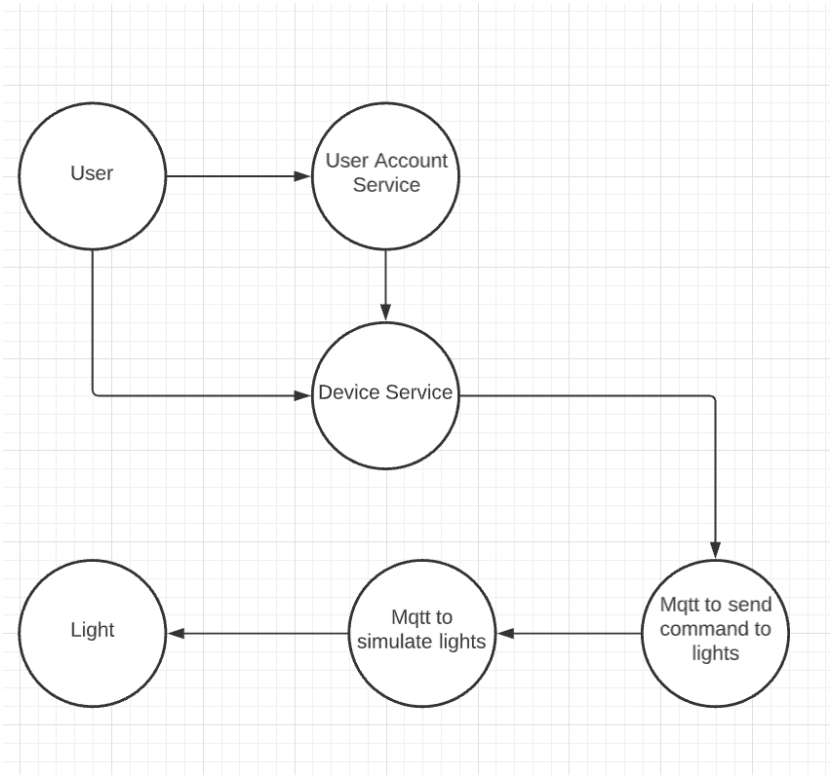
- To have a centralized smart lighting controller system, we need to host a web server online and a web Api. We should be able to connect our motion sensors to switch lights On/Off by posting to a certain API endpoint. We can use an array of motion sensor and lights in the building and connect online maybe we can use a Raspberry Pi. We can have an automatic function which will make the system switch off all lights when automatically during the day and during sleep time in the night.
- Not many smart lighting systems have any kind of security and have good GUI-control systems as well. Also, we need to ensure that our system is scalable quite easily. We cannot use local storage and computation to host our severs online as we might not have sufficient resources to handle the load of the system when more and more lights are added to the building or more features are added to it.

# Methodology

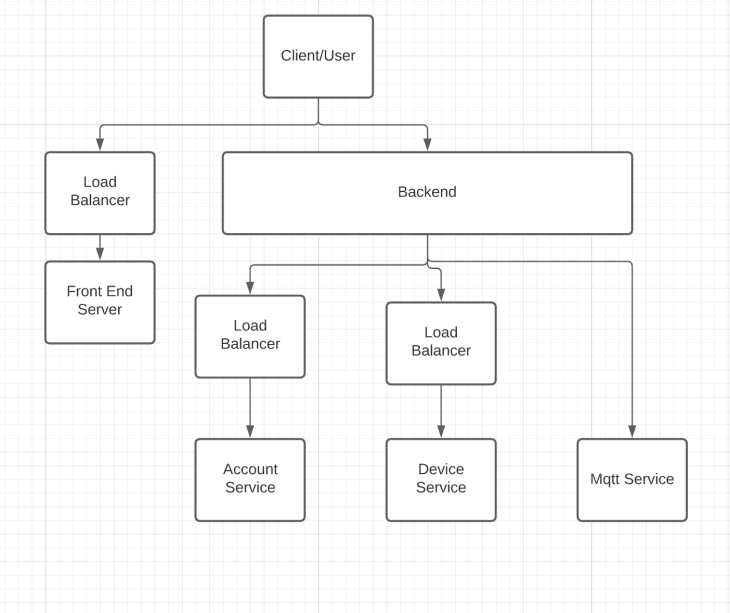
# Mapping all of the business functions and their connections in the form of an informal diagram



# Microservices



# System Design



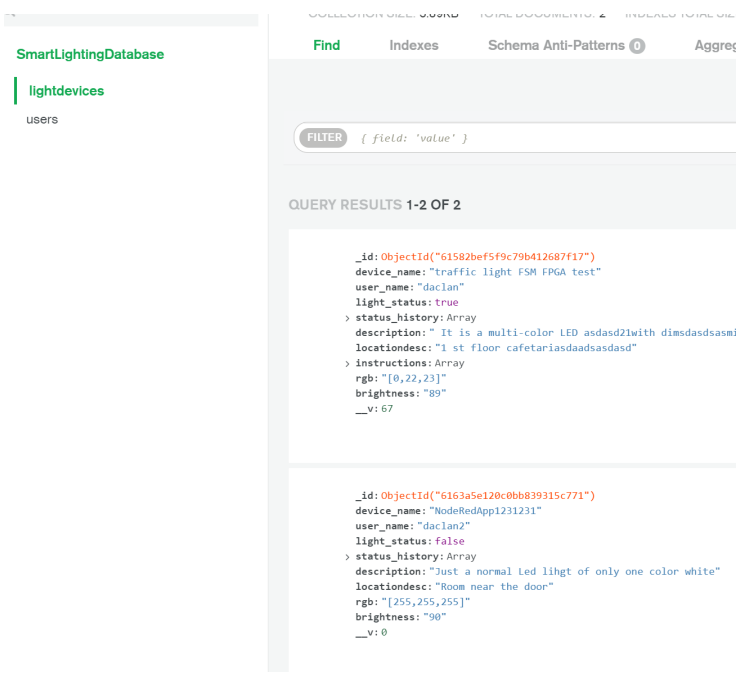
# Demonstration with some insights

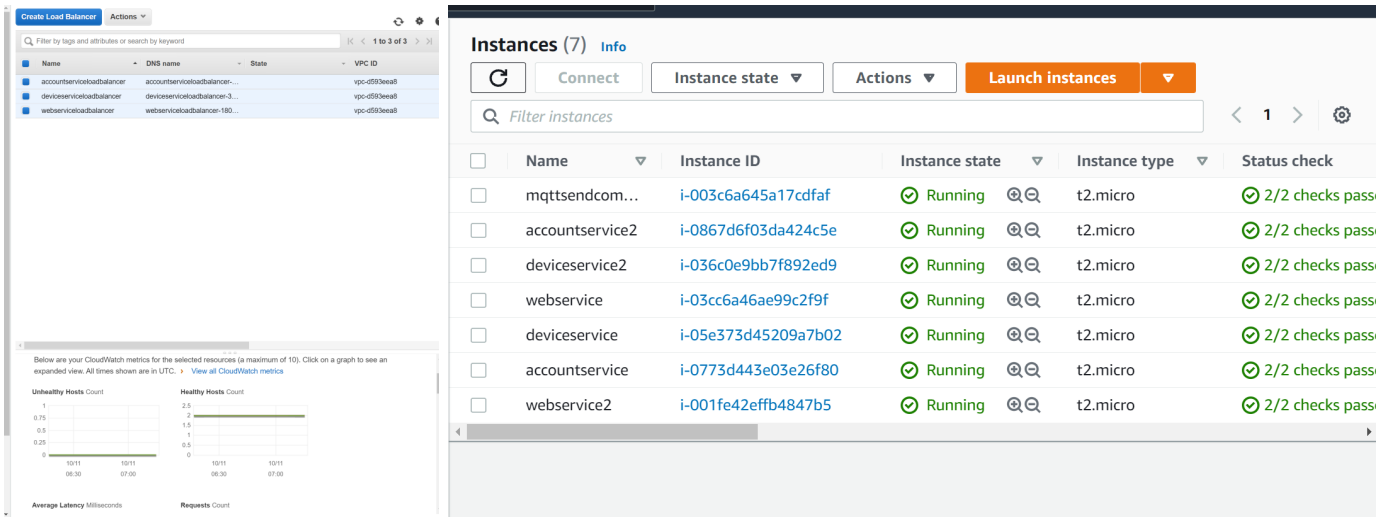
# Data formats

```
const mongoose = require('mongoose');  1006.9K (gzipped: 272.8K)
module.exports = mongoose.model('User', new mongoose.Schema({
name: String,
password: String,
isAdmin: Boolean,
notification_array:Array,
email_id:String
}));
```

```
services > mqtt light simulator service > models > JS lightdevice.js > ...
const mongoose = require('mongoose');  1006.9K (gzipped: 272.8K)
module.exports = mongoose.model('LightDevice', new mongoose.Schema({
id: String,
device_name: String,
user_name: String,
light_status:Boolean,
status_history:Array,
description:String,
locationdesc:String,
rgb:String,
brightness:String
}));
```

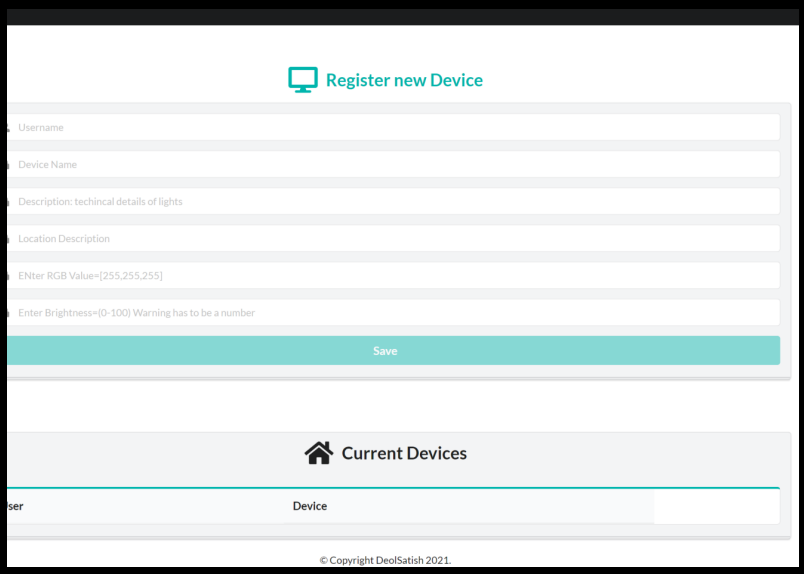
# MongoDB database

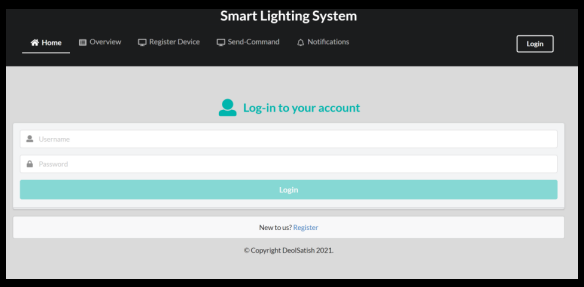


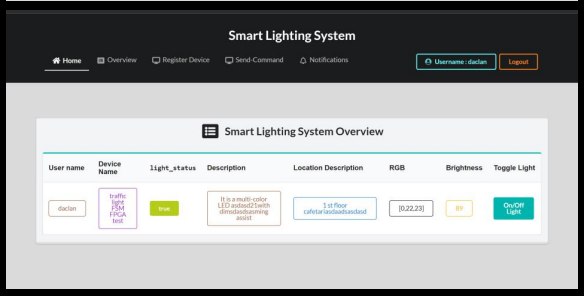



# AWS EC2 Instance and Load-Balancers

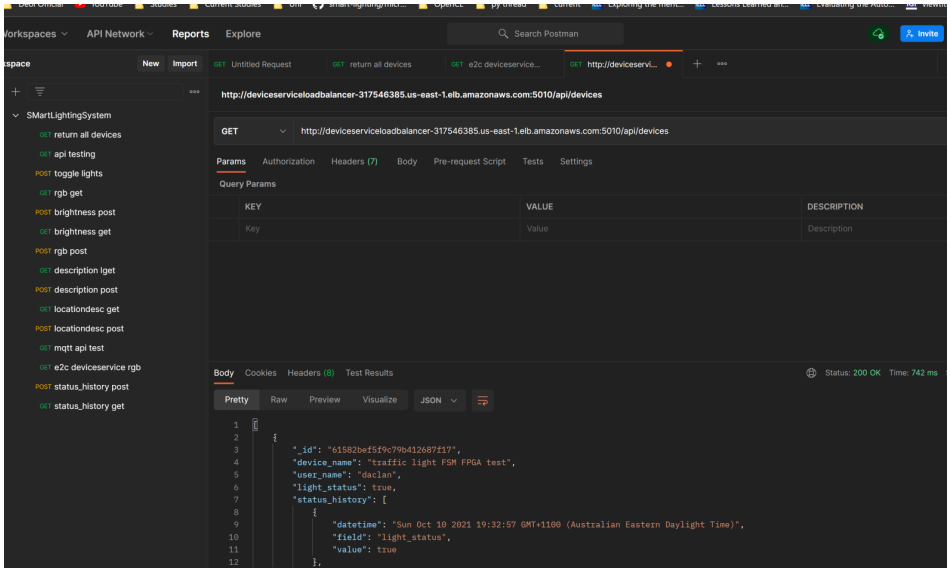
# Web-UI Interface



# Microservices Endpoints



# Security Features

- It requires a user to login to access and control the light devices
- The system uses private Hivemqtt cluster and is required to have a username and password to subscribe to this cluster.
- More security features we can add:
  - We can add jwt token authorization for api endpoints
  - We can add Oauth authentication with using google, facebook account etc
  - We can add 2-factor authentication, where a pin code is sent to the user's email-Id

# Issues faced while prototyping

- Issues regarding the microservices and the UI service was no that difficult as we already have a bit of experience in web application development.
- Major problems were caused due to the load balancer and when hosting the web service microservice on an e2c instance.
- The loadbalancer problem was caused due to unhealthy checks, because I had put the wrong ping path, which took me hours to realize what I did wrong
- The other problem was with the frontend not working when It automtaclly start the server using crontab –e. So, I used another module named pm2 to fix this issue. It works in a similar fashion to crontab –e but it is much easier to use.

# Conclusion

- I think the prototyping has just started and there are a lot more things we can do, especially using real hardware. I think the prototyping went well, there are a few more features I wanted to have, but time did not permit.
- Truthfully speaking, I love doing the microservices, the mongoDB database connection, setting up AWS instacnes and load balancers.
- Basically everything related the functioning of the system was a lot of fun. I did not like doing the GUI that much. I have inherent problems with designing and stuff. It is hard foe me to pick decide stuff especially when there is no logical answer.

Link to video:https://youtu.be/LSvEYNGFgeU

https://video.deakin.edu.au/media/t/1_e9381js1