REST API for a Wireless Sensor Network

BY: SAKIB CHUGHTAI

PROJECT SUPERVISOR: DR KAMYAR MEHRAN

20-8-2019

Presentation Structure

- Problem Statement and Justification
- Aims and Objectives
- Related work and Literature
- Conceptual Frameworks
- Methodology
- Findings
- Conclusion
- Implications and Recommendations

Problem Statement & Justification

- Monitoring energy consumption has become increasingly important as a result of rising energy costs.
- Whilst Wireless Sensor Networks enable monitoring, managing sensor networks requires custom APIs which is tedious and error prone.
- Justification
  - Use REST framework to ease the process of interacting with sensors and making them accessible over the web.
  - Rise in popularity of Smart Home Technology and Internet of Things

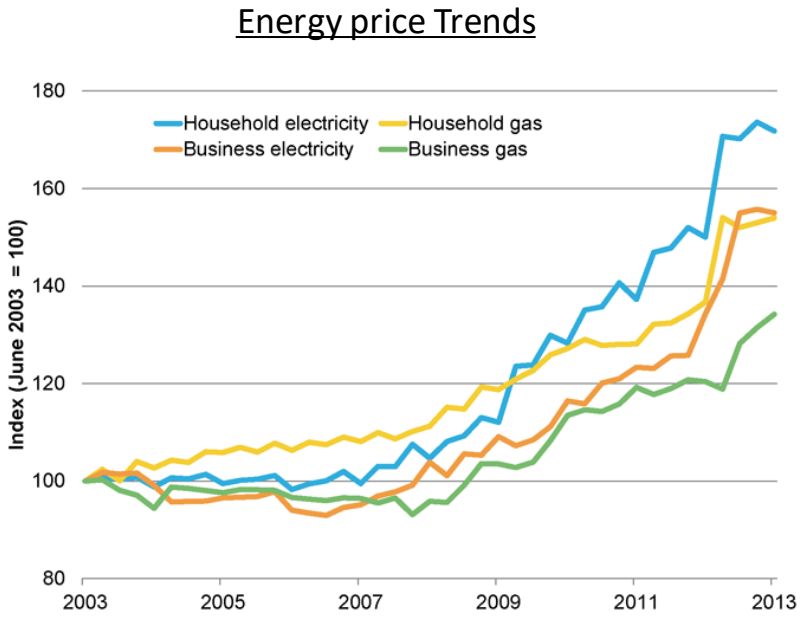


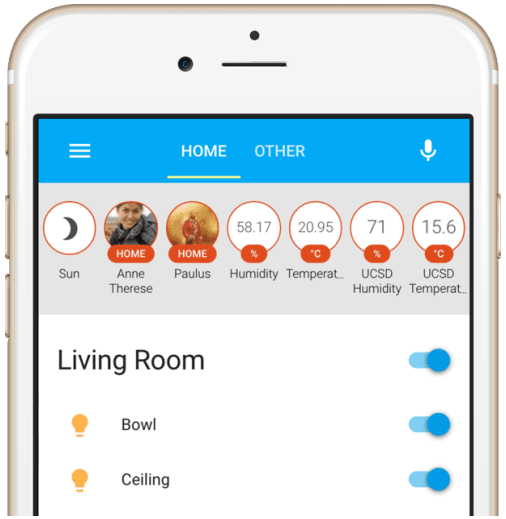
Figure 1: showing trends in energy prices across households and businesses

Aims & Objectives

- Aims
  - To provide users with insights on the energy consumption of the appliances they use in their household.
  - To provide users with an estimated cost of energy for appliances within the household, so they can make smarter decisions on how to reduce energy costs.
- Objectives
  - Implement an efficient and scalable web application that allows users to monitor energy consumption of appliances and provide an estimated cost.
  - Users should be able to view all/specific appliances in their household.

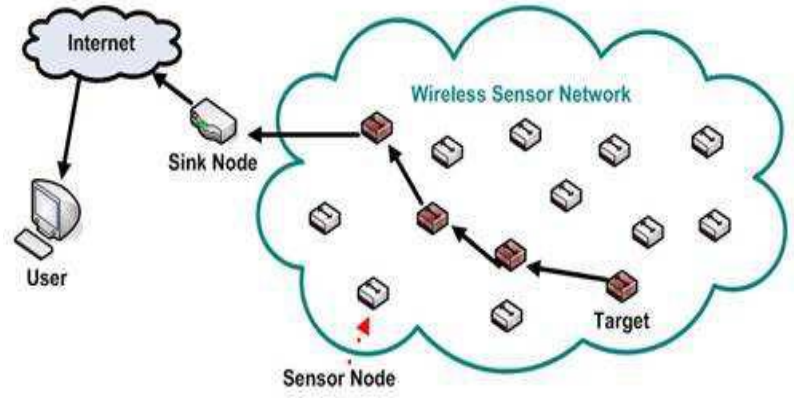
Related Work & Literature

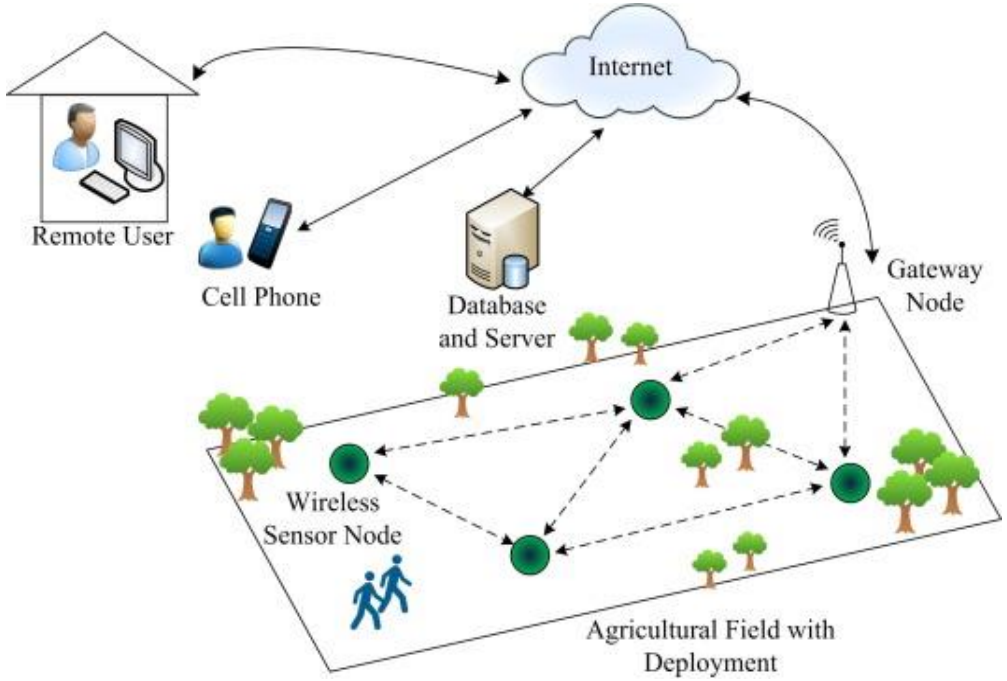
- Smart Home Technology
- Internet of Things
- Home Assistant
- Literature Review
  - WSN: A Betts
  - REST: Roy Fielding, A Diaz



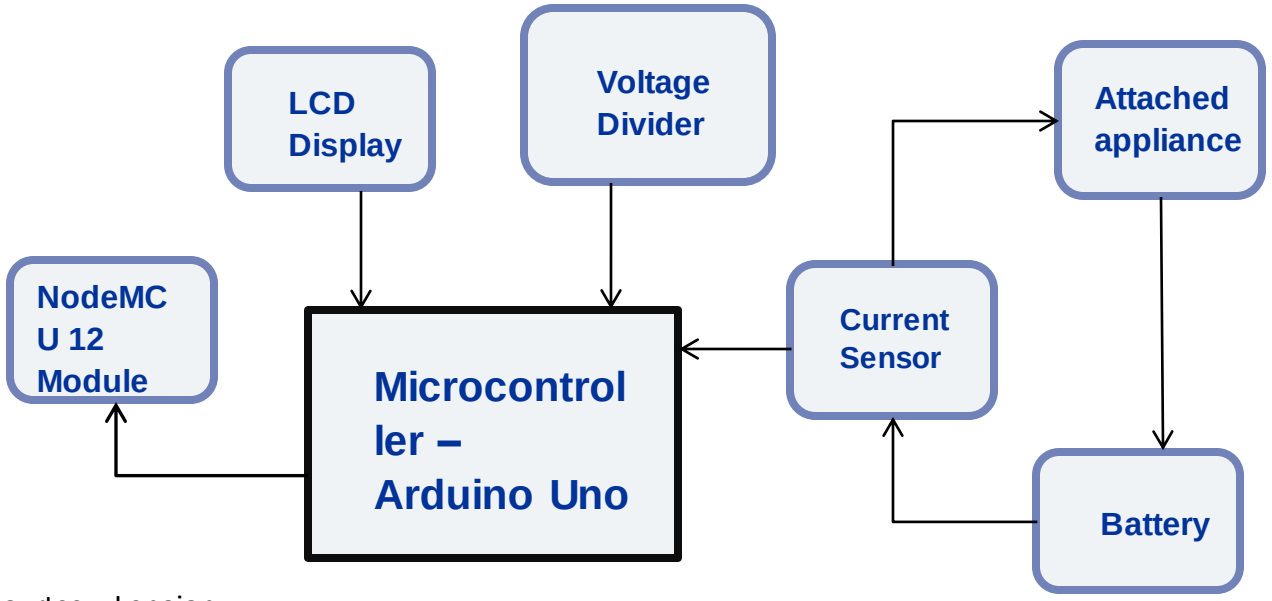
Conceptual Framework: Wireless Sensor Network

- A system which that wirelessly monitors physical and environmental properties through the use of sensor nodes.

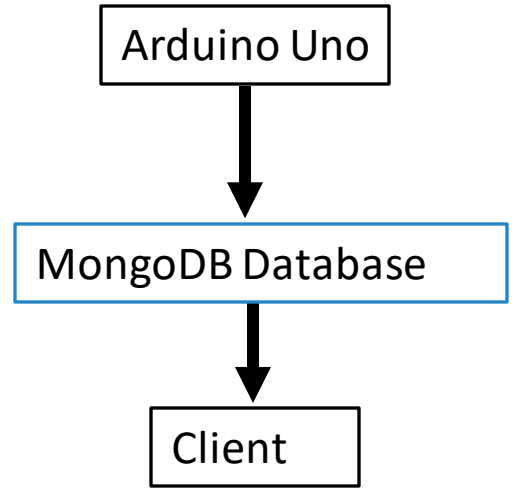




Hardware & Flow of Data

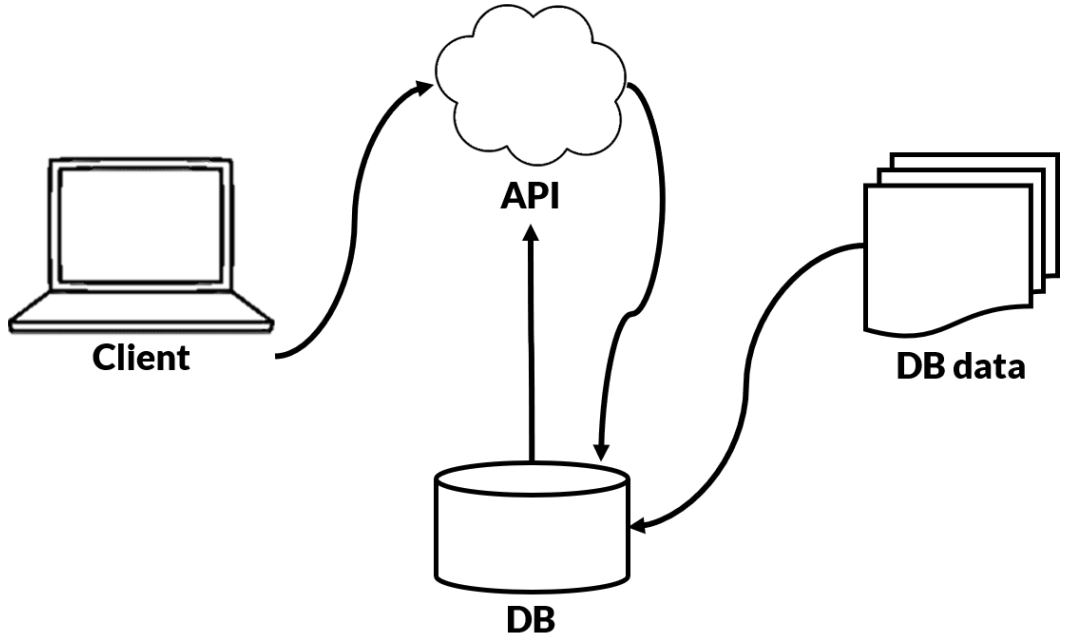


Courtesy: Lenojan Jeyarajan



Conceptual Framework: REST API

- Representation State Transfer
- Commonly used for web applications.
- Main reasons for choice:
  - Stateless architecture
  - Uniform interface
  - Layered system
  - Provides potential for scalability
- 4 principles used to interact with data:
  - GET
  - POST
  - PUT
  - DELETE

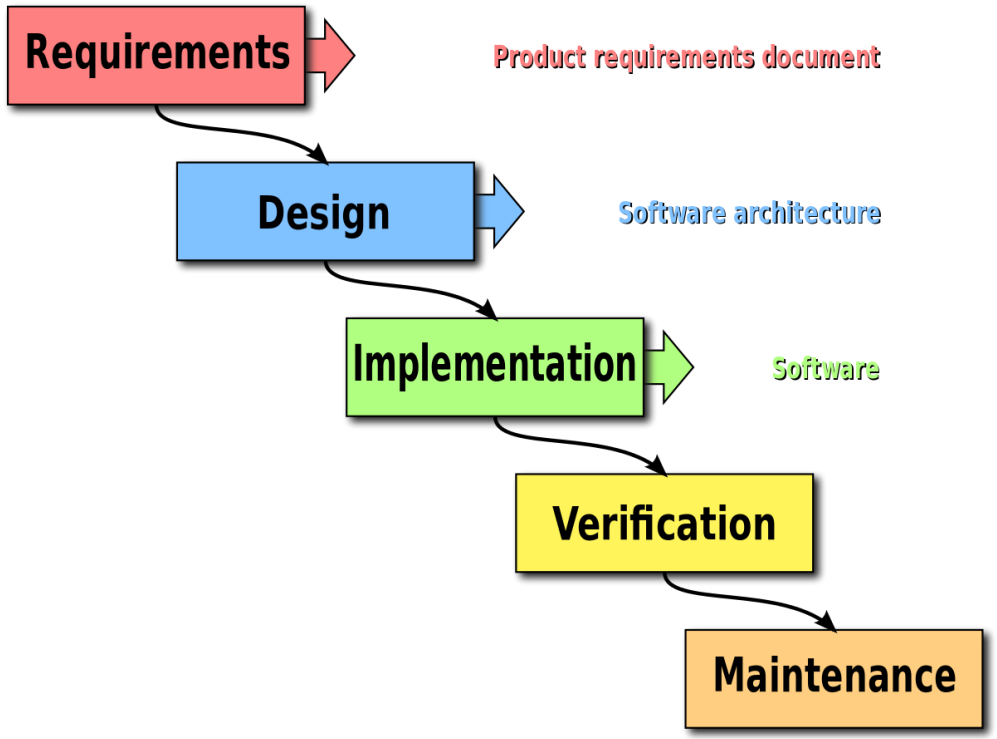


Important Technologies used

- Node.js
  - Open source JavaScript runtime used to build network applications.
  - Provides “full-stack” capability for client and server side operations.
- Express.js
  - Enables definition of routes between application endpoints and requests from the client.
  - Composed of Routes, Router and Middleware.
- MongoDB
  - Non-relational database that collects are stores documents formatted in JSON.
  - Hosted in the cloud for scalability and security purposes.

Methodology

- Waterfall development method
  - Sequential design process, commonly used for many software implementations.
- Data analysis & performance measurements
  - Load Testing for each REST component.
  - Network Usage.
  - PC usage (memory, CPU).
- Non functional/functional Requirements
  - Users must be able to view all appliances in the database.
  - Each GET request must be processed by the application within 100ms.



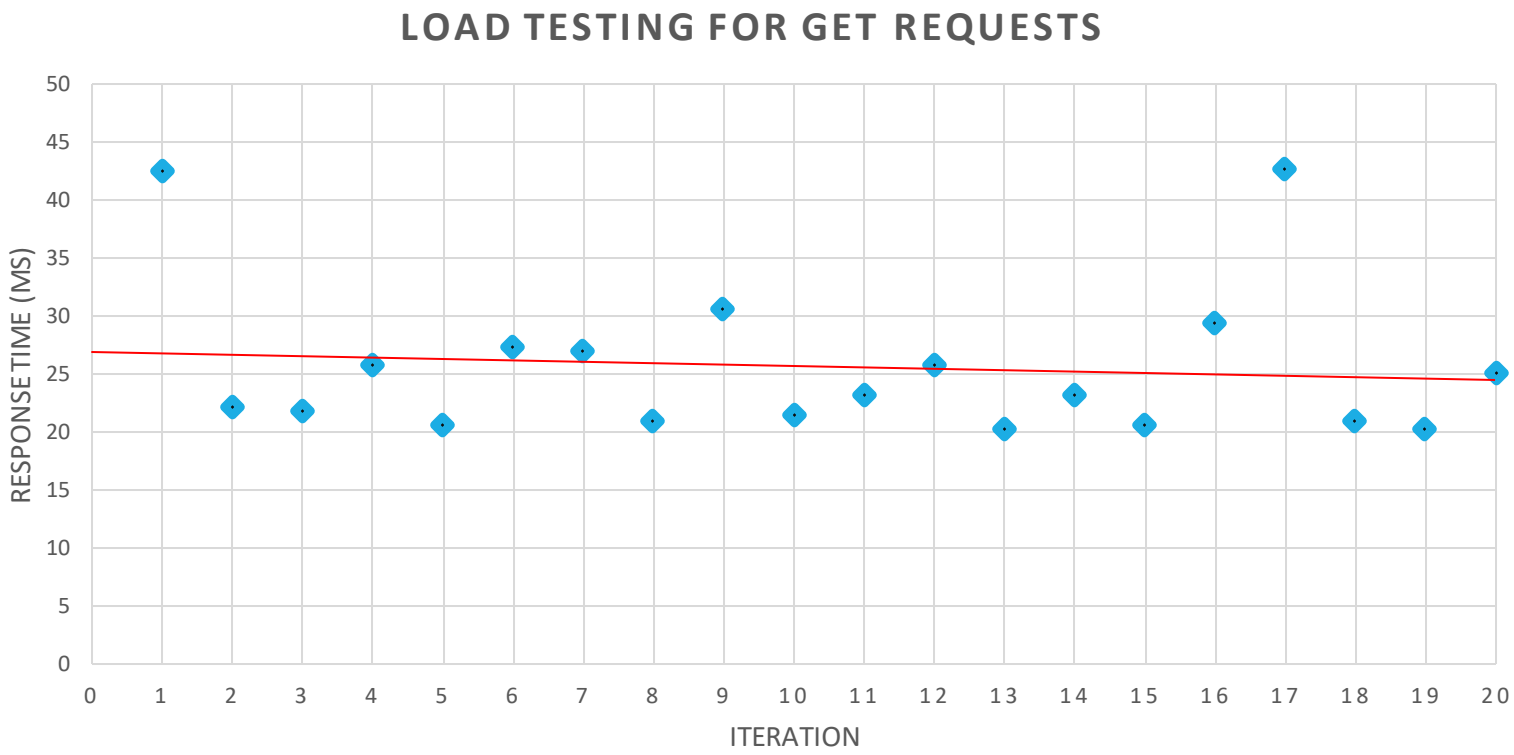
Methodology: Sensor Model

- Schema – defines how each sensor is modelled and stored within the database.
- Essential component of the whole system.

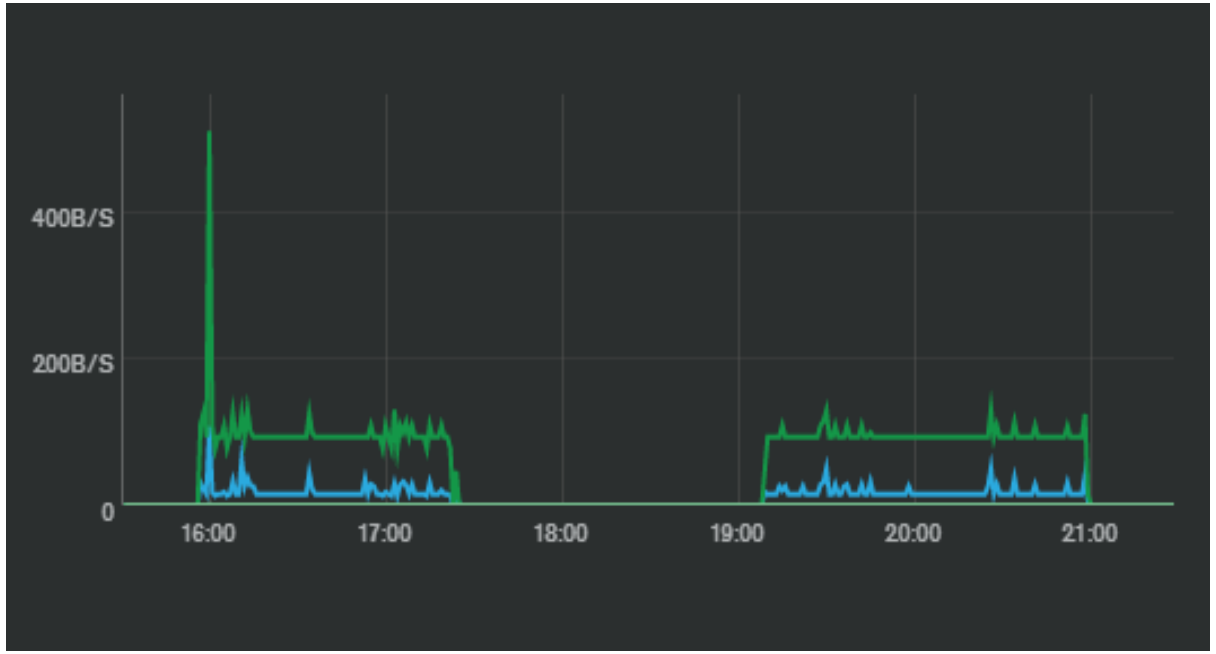
```
10  var senSchema = mongo.Schema({
11      _id: mongo.Schema.Types.ObjectId,
12      name: String,
13      voltage: Number,
14      watt: Number,
15      costperHour: Number,
16      costperDay: Number,
17      costperMonth: Number,
18  });
```

Findings: Load Testing

Avg Response Time = **25ms**



Findings: Network Usage



Graph showing network usage (Bytes per second) for two different time intervals, where green = bytes in, blue = bytes out.

Findings: Data Retrieval

Example of retrieval of data.


```
_id: ObjectId("5cbf8b41c04ad54be4722eba")
name: "Microwave"
voltage: 5.76
amps: 22
costperHour: 0.034
costperDay: 0.82
costperMonth: 24.48
__v: 0
```

```
_id: ObjectId("5cbfec3f77f3bc4b2869382d")
name: "Kettle"
voltage: 10.2
watt: 2000
costperHour: 0.052
costperDay: 1.25
costperMonth: 37.44
__v: 0
```

```
_id: ObjectId("5cbfed0777f3bc4b2869382e")
name: "TV"
voltage: 110
watt: 150
costperHour: 0.042
costperDay: 1.08
costperMonth: 30.24
__v: 0
```

Conclusion

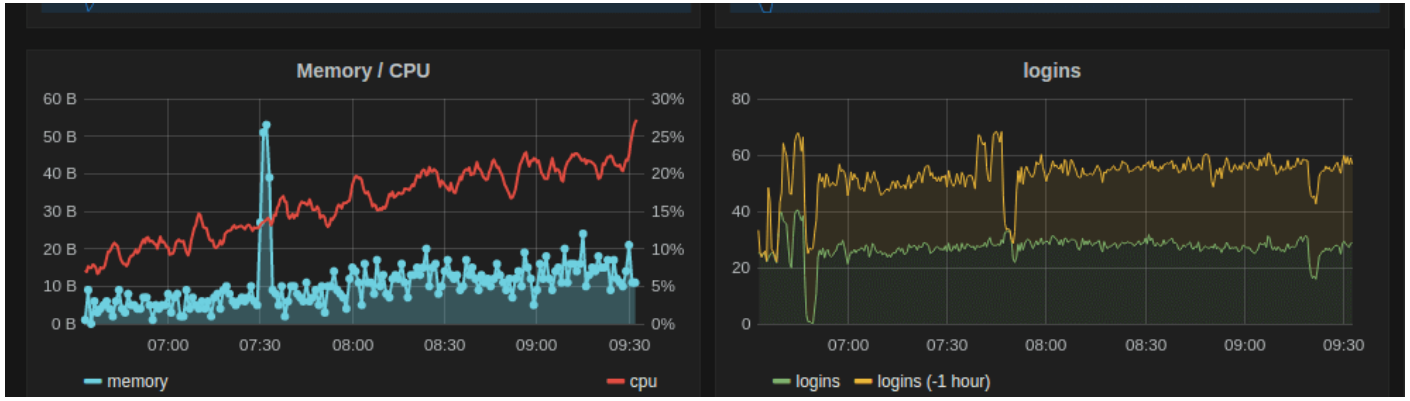
- REST API provides the most efficient solution for managing interactions between a client and database.
- The combination of Node.js, REST and MongoDB provides an effective method of creating a useful web application, in one language (JavaScript).
- Load testing highlights that the application is performing better than expected, and is capable of handling large volumes of data (Scalability).
- Fast response times indicate that the applications performs well, which is one of the fundamental requirements of the software.

Strengths and Weaknesses

- Strengths
  - Web application is fast and efficient.
  - No bottleneck issues.
  - Fast response time.
  - Data is secured throughout each request/response.
- Weaknesses
  - No recommendations based on trends in energy consumption.

Implications & Recommendations

- Ethical implications
- Recommendations for future research
  - Machine Learning: Applying algorithms to collected data to provide insights into energy consumption.
  - Visual Representation of data.
  - Enhancing Security: Using HTTPS instead of HTTP protocol.



References

Anon, (2016). *REST API: What is it, and what are its advantages in project development?*. [online] Available at: https://bbvaopen4u.com/en/actualidad/rest-api-what-it-and-what-are-its-advantages-project-development [Accessed 1 Jan. 2019].

Anon, (2019). *Introduction to Node.js*. [online] Available at: https://nodejs.dev/ [Accessed 19 Feb. 2019].

Caccavale, M. (2018). *Council Post: The Impact Of The Digital Revolution On The Smart Home Industry*. [online] Forbes.com. Available at: https://www.forbes.com/sites/forbesagencycouncil/2018/09/24/the-impact-of-the-digital-revolution-on-the-smart-home-industry/#263f06483c76 [Accessed 29 Jan. 2019].

Docs.mongodb.com. (2019). *Read Data from MongoDB With Queries*. [online] Available at: https://docs.mongodb.com/guides/server/read_queries/ [Accessed 17 Feb. 2019].

Docs.mongodb.com. (2019). *TLS/SSL (Transport Encryption) — MongoDB Manual*. [online] Available at: https://docs.mongodb.com/manual/core/security-transport-encryption/ [Accessed 12 Jan. 2019].

Expressjs.com. (2019). *Express routing*. [online] Available at: https://expressjs.com/en/guide/routing.html [Accessed 10 Jan. 2019].

Expressjs.com. (2019). *Writing middleware for use in Express apps*. [online] Available at: https://expressjs.com/en/guide/writing-middleware.html [Accessed 15 Feb. 2019].

https://www.pinterest.co.uk/pin/324188873161328578/?lp=true

https://medium.com/@ashish_fagna/influxdb-to-grafana-visualizing-time-series-data-in-real-time-2174940a894d

https://www.aliexpress.com/item/Sonoff-Pow-R2-16A-Power-Energy-Meter-Monitor-Wireless-WiFi-Switch-with-Timing-Sharing-Function-Remote/32863611839.html

https://www.johnlewis.com/amazon-echo-smart-speaker-with-alexa-voice-recognition-control-2nd-generation/p3348223

- Fredrich, T. (2018). *HTTP Methods for RESTful Services*. [online] Restapitutorial.com. Available at: https://www.restapitutorial.com/lessons/httpmethods.html [Accessed 14 Feb. 2019].
- Home Assistant. (2019). *Hass.io*. [online] Available at: https://www.home-assistant.io/hassio/ [Accessed 15 Jan. 2019].
- Kocakulak, M. (2017). *An overview of Wireless Sensor Networks towards internet of things - IEEE Conference Publication*. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/7868374 [Accessed 8 Jan. 2019].
- Matin, M. (2012). *Overview of Wireless Sensor Network*. [online] Intechopen. Available at: https://www.intechopen.com/books/wireless-sensor-networks-technology-and-protocols/overview-of-wireless-sensor-network [Accessed 15 Jan. 2019].
- Medium. (2017). *Why Would You Use Node.js*. [online] Available at: https://medium.com/the-node-js-collection/why-the-hell-would-you-use-node-js-4b053b94ab8e [Accessed 16 Feb. 2019].
- Node.js. (2019). *Overview of Blocking vs Non-Blocking | Node.js*. [online] Available at: https://nodejs.org/en/docs/guides/blocking-vs-non-blocking/ [Accessed 11 Jan. 2019].
- Parsons, N. (2017). *MongoDB Atlas — Technical Overview & Benefits*. [online] Medium. Available at: https://medium.com/@nparsons08/mongodb-atlas-technical-overview-benefits-9e4cff27a75e [Accessed 12 Feb. 2019].
- Singhal, R. (2018). *Restful API Design*. [online] Medium. Available at: https://medium.com/@rachna3singhal/restful-api-design-95b4a8630c26 [Accessed 12 Jan. 2019].
- w3resource. (2018). *MongoDb Tutorial - w3resource*. [online] Available at: https://www.w3resource.com/mongodb/introduction-mongodb.php [Accessed 22 Dec. 2018].
- En.wikipedia.org. (2019). *Waterfall model*. [online] Available at: https://en.wikipedia.org/wiki/Waterfall_model [Accessed 16 Aug. 2019].
- Ojha, T. (2019). *Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges*. [online] Science Direct. Available at: https://www.sciencedirect.com/science/article/pii/S0168169915002379 [Accessed 11 Aug. 2019].
- UK, P., System, S., Gordon, W. and Gordon, W. (2019). *How to Set Up Your Smart Home: A Beginner's Guide*. [online] PCMag UK. Available at: https://uk.pcmag.com/simplisafe-home-security-system/120237/how-to-set-up-your-smart-home-a-beginners-guide [Accessed 16 Aug. 2019].