INFT3970 Major Project Scope Document
Distributed Monitoring System using Embedded Devices

Team Encore
Jay Rovacsek c3146220@uon.edu.au
Dean Morton c3252227@uon.edu.au
Josh Brown c3283797@uon.edu.au
Jacob Litherland c3263482@uon.edu.au
Lee Marron c3263482@uon.edu.au
Edward Lonsdale c3252144@uon.edu.au

August 23, 2018

Contents

1 Executive Summary

1.1 Background

Riding the wave of "IoT Revolution" [1], this project will develop a low cost, easily deployable IoT product in any setting.
IoT or *The Internet of Things* has proven to be an explosive trend within the consumer electronics markets. Never before have such small versatile devices been available for general consumption, leading to an estimated combined business and consumer spending value in excess of $6 *trillion* dollars globally in 2018 [1].
For the vast majority of consumers, both corporate and end-customer [2], a large movement toward both minimisation of waste and optimisation of spending is occuring on a global scale in the developed world, with the developing world rapidly following this trend also [3].
Citing this movement, it would only be logical to create a simple to use set of devices that allow for the monitoring and therefore optimisation of such measurables.

1.2 Overview and Purpose

The concept of this project is to create a distributed system in which small devices are used to monitor, log and analyse a number of select metrics from a multitude of potential data points.
The purpose of the project is to deliver a viable product that could be replicated for a reasonable price for both end-user and business alike. We believe the market to be on a precipice of further explosive growth, with the consumer market partially realised, but far from tapped by curent offerings.
In this document we intend

1.2.1 Metrics

The metrics measured included will be:

- Temperature
- Humidity
- Motion

We anticipate further development on the project to be viable post submission date, however realise the limitations of the current timeframe.

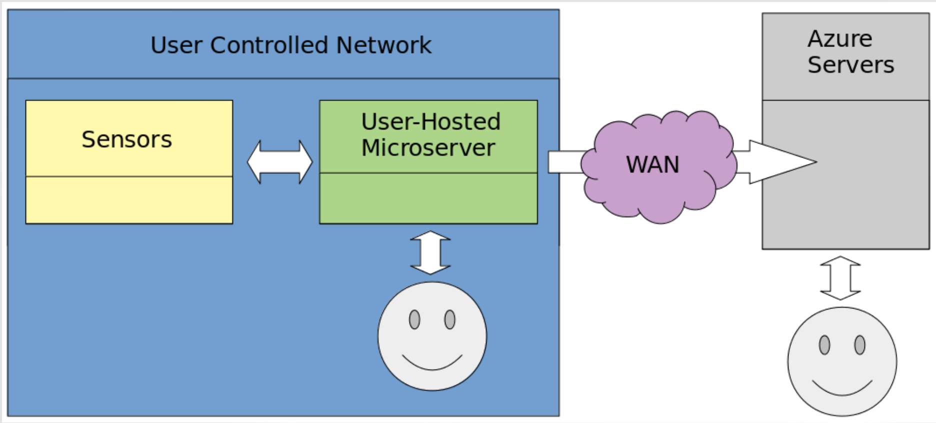
Metrics measured would be viewable on a users dashboard, with data being able to be scoped to multiple filter requirements such as time, select edge cases or specific locations.
The end-goal being an ability for users to better determine inefficient or bad decisions they may make unwittingly in regards to home or business heating, coupled with the impact of room utilisation.

2 Business Needs

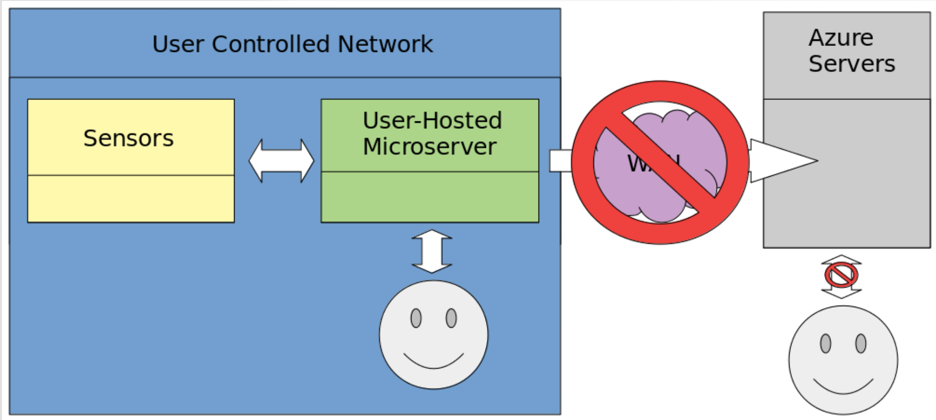
As suggested in the executive summary and project background, the IoT market is expected to rapidly grow in not only the current year but a number of future years to come. Accessibility and ease of use being major factors spurring the market into the current state.
This project aims to fill some of the gaps that exist within the expanding world of IoT by allowing end users irrespective of technological capability to easily partake in the current market at a price position that doesn't reflect excessive or overly expensive alternatives that exist.

2.1 Azure Hosting

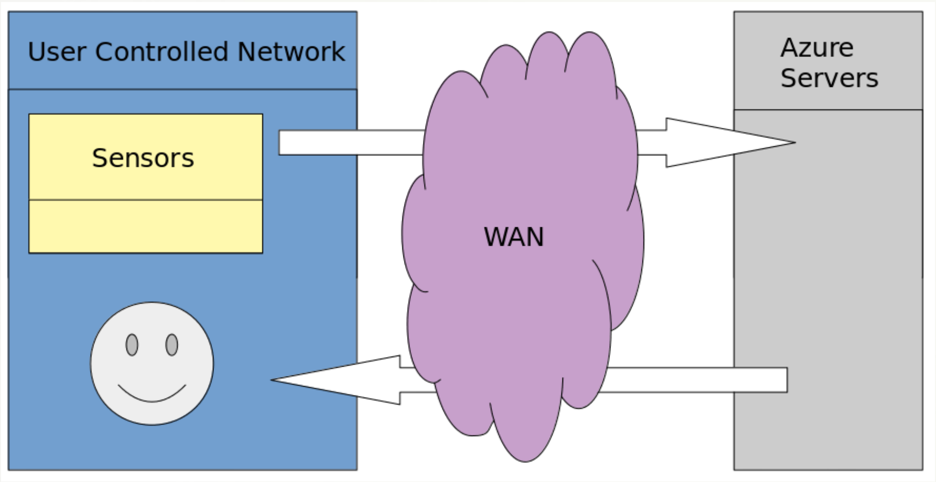
Currently without extenuating circumstances against hosting within the cloud the decision to host in Azure applications seems to be the path of least resistance for the project in order to enable a globalised access to a single database to host data.
We recognise the flaws with hosting in the cloud also, such as data breach potential (traditional methods or emerging threats [15]), however also recognise with a limited span of time allowable for the project a more suitable system design such as below is not suitable in the timeframe:



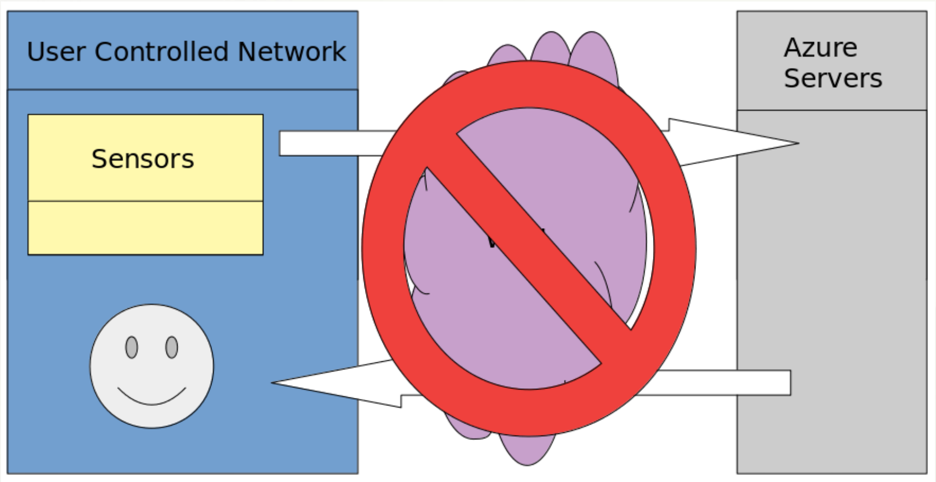
Such a design would allow for users to maintain connectivity from their own LAN:



Given time constraints we need to accept that the design of centralised servers hosted within Azure achieves the same ability to serve the user information and maintain connectivity as required, but is more failable:



In a network failure situation:



Optimally the scope of the project would be expanded to development of the user hosted microservers that would act as middleware between the hosted servers and the sensors, this is a highly viable goal given technologies such as the RaspberryPi [13] which under $50AUD can a number of Linux distributions or Windows IoT Core platform.

2.2 WLAN v Zigbee

Considerations to implement the network of devices was considered however given the requirement to purchase and implement a module to enable use of Zigbee, conventional WEP/WPA/WPA2 network options were perfered. Both conventional WLAN networks and Zigbee would allow meshability between devices, requiring less of an impact on the user's network in a mixture of DHCP leases and wireless network congestion.

3 Project Objectives

Within the timeframe still available to this project, we aim to develop and deploy a number of IoT devices [4] to a home enviroment or two and to track heat, humidity and motion of the dwelling to better understand the potential correlations of room use, heating and potential inefficiencies created in areas such as 'High Traffic' spots (Loungerooms and Hallways)
Optimally we ain to couple this with a mapping of the dwelling, allowing a more intuitive expression of the data collected.

We intend on using student subscriptions to leverage Azure for both website hosting and databasing coupled with a small budget of roughly $100-$200 to purchase all required equipment which currently is expected to be:

- Arduino UNO3 Microcontroller [10]
- ESP8266 boards [4]
- DHT11 Temperature and Humidity sensors [6]
- XC-4444 PIR Motion sensors [5]
- Various required breadboards / generic electronics items

The overall goals of the project is to enhance our knowledge of of IOT sensor technology in the scope of exploratoring what this technology can achieve and expanding the paradigm of this technologies implementations.

Secondarily the group seeks to develop our ability to interface with middleware using a number of appropriate languages to create a frontend user interface for users to monitor/operate and perform data analytics on the system and a backend for data management system that interfaces with an SQL server implementation.

We intend in this project to showcase out abilities while expanding our capabilities and have an aim to deliver a prototype by week 9.

4 Deliverables

The deliverables that are expected to be produced throughout the distributed monitoring system are broken into Four sections; Sensors, Database, UI and analytics.

4.0.1 Sensors

- Humidity Sensor can be created.
- Temperature Sensor can be created.
- Motion Sensor can be created.
- Sensors can be reprogrammed with different modules (Or have all modules avilable).
- Sensors ownership can be modified.
- Sensors cav connect to any 2.4gHz home network.

4.1 Database

- A new user account can be created.
- A new sensor can be registered.
- An admin can access user's information (password hashed + salted)
- A user record can be edited.
- A sensor can be assigned to a user account.
- A sensor can be deleted from a user's account.
- A sensor's details can be edited.
- An admin can view raw data from all tables.
- An admin can edit data from tables.
- Password can be changed
- Email will be sent for resetting a forgotten password
- Email verification upon account creation
- Procedures and triggers for data analysis and querying

4.2 User Interface

- Login page for Users.
- Registration page for Users.
- Home page (Displaying an overview).
- Temperature page.
- Humidity page.
- Motion page.
- User able to access associated data.
- Graphs based on the user data.
- User able to customise graphs and Home page layout.
- Custom Analytic results.
- Email Suggestions section.
- Logout Page.

4.3 Data Analytics

- Administration able to view user data that has been depersonalised.
- Display Recommendations and habits to the User.
- Showing different trends around a user's house using the different modules.

Having a Data Analytics major within our group we expect the data collected may contain interesting information.
An analytic module will be used to parse data for each user and display recommendations and habits from the data.

4.3.1 Temperature Module

The analytic module will be able to tell the users how different the temperature is around their house and see trends on room temperatures. This will also be able to be used to give recommendations on what rooms need heating/cooling and what rooms do not benefit from heating or cooling.

4.3.2 Humidity Module

The analytic module will be able search through the data and view what rooms and areas of the house are the most humid. This allows the user to keep an eye on the moisture that is in their homes like a wardrobe or kitchen pantry. Another use for the humidity module is alongside the temperature module to give suggestions to the user if they should turn on/off Air-conditioning units.

4.3.3 Motion Module

The analytic module will be able to see trends and habits on behaviour as they move through their home. By using this information the user will be able to see what rooms are high traffic areas and at what times of the day this high concentration of movement occurs. Through this it can also be used to tell when outliers occur for security purposes. Such as late night movement in a shed could be an intruder and the user will be notified with this uncommon event.

4.4 Milestones

A number of highly important milestones have been already scheduled as below;

Milestone	Date
Brainstormed viable projects, finalised project choice	12th August
Project broken down into milestones, assigned initial work to members	14th August
Initial database and sensors created	19th August
Sensors connecting to the database	20th August
Finished Scope Document	22nd August
Working User interface prototype	9th September
Programming complete for all sensors	16th September
Completion of Database and scripts	1st October
Finished User Interface	14th October
Testing completed	20th October
Project Completed for Exhibition	25th October

Otherwise milestones for the project include are split into a number of concurrently developed subsections:

1. Database:
    1.1. Database Pilot (Including creation and design)
    1.2. Review and Testing of Database
    1.3. Optimisation of datatypes used in each table
    1.4. Implementation of indexing and other optimisations to avoid big O issues.
    1.5. Final database implementation.
2. Backend:
    2.1. Decision of framework to be used
    2.2. Initial POC on framework
    2.3. Feature implementations:
        2.3.1. Implementation of PING GET endpoint to avoid sending of data unnecessarily
        2.3.2. Implementation and testing of POST endpoints:
            2.3.2.1. Temperature
            2.3.2.2. Humidity
            2.3.2.3. Motion
        2.3.3. Implementation and testing of application helpers to aid front-end data sourcing
    2.4. Refactor existing code
        2.4.1. Peer review of code, implementation of required changes
3. Front End:
    3.1. Initial implementation of POC dashboard
    3.2. Initial POC of authentication using a login page
    3.3. Review of code by peers
    3.4. Migrate dashboard behind successful login page
    3.5. Improvements and refactoring of dashboarding code
    3.6. Review and refactor to suit all devices commonly used to browse websites.
    3.7. Implement required bootstrap/css framework to handle display of data.
    3.8. Final peer review and refactor

4. Sensors:
    4.1. Pilot testing of potential boards (UNO3 [10] / ESP [4] / RPI [13] / OPI [14])
        4.1.1. Code feasability report on each board, considerations on support from community and team ability
        4.1.2. Review of boards and decision on main boards to use
    4.2. Implement monitoring for chosen metrics:
        4.2.1. Temperature
        4.2.2. Humidity
        4.2.3. Motion
    4.3. Implement networking on boards
        4.3.1. Determine viability of WPA2-Enterprise connectivity
    4.4. Implement json generation on boards
    4.5. Implement PING request functionality to check web server is up or not
    4.6. Test API communication
    4.7. Design and print 3D case for boards
    4.8. Deploy boards to various locations
5. Documentation
    5.1. Generate POC documentation from project
    5.2. Generate scope document
    5.3. Comment code extensively for readability
    5.4. Document APIs
    5.5. Document database design, with considerations of optimisation and required modifications
    5.6. Generate user guide for final product
    5.7. Generate final report

5 Key Roles in Project

Key roles defined for the project include the below:

Task	Task Manager	Support
Brainstormed viable projects, finalised project choice	Whole Team	Whole Team
Project broken down into milestones, assigned initial work to members	Jay	Whole Team
Initial database and sensors created	Lee	Whole Team
Sensors connecting to the database	Jay	Dean, Lee
Finished Scope Document	Jay, Lee	Whole Team
Working User interface prototype	Ed	Jacob, Lee, Josh
Programming complete for all sensors	Jay	Dean, Lee
Completion of Database and scripts	Josh	Ed, Jacob
Finished User Interface	Ed	Jacob, Lee, Josh
Testing complete	Dean	Whole Team
Project completed for Exhibition	Whole Team	Whole Team

6 Technical Requirements

6.1 Generalised Requirements

The team collaboratively have agreed on a few generalised requirements of the project including:

- Ease of use: The devices must be able to used by a nontechnical user, they must be supplied in such a way that an ability to place the device in place, and connect power are the largest difficulties that the end user encounters.
- Abstraction of underlying technologies: as a byproduct of the requirement of ease-of-use the devices and web application should not require the end user to have any knowledge of the underlying technolgies used.

6.2 Data Requirements

Requirements for raw data be stored in an online database are only access for the development resources to be able to access any suitable cloud account, self-host on a local machine or debug required functions in the application with localised data generation via ajax [17] queries of similar methods.

6.2.1 Azure

Azure is a likely candidate for the Database and website hosting, The database will be used to store all the data that is sent from the sensors. The uptime of the database the web application is vital to the user experience and the success of the project. Azure provides this security with 99.95% uptime [9].

6.3 Hosting Enviroment

Pages for the site itself will require a suitable host either cloud or local which supports .NETcore 2.0 [7] and suitable network access to reach a required TSQL [8] database hosted by the team.

6.4 Network Connectivity

Functioning network connectivity will be heavily depended on by the project, all of the sensors will be connected to a local network wirelessly which will enable the boards to post data at the API endpoint at either `https://www.inft3970.com` (Parked domain) or `https://inft3970.azurewebsites.net` (Currently functioning endpoint)

Requirements for the user to maintain a 2.4gHz enviroment will exist, as without this an expansion board or alternate boards will be required isntead of the proposed ESP8266 [4] and UNO3 [10].

6.5 User Requirements

The users of the system will need to be confident in browsing webpages and accessing them. Generally we recognise any device with browser functionality will suit the purpose, and development will use a mixture of Firefox [11] or Chromium [12] based browsers to find all possible bugs within the project span.
The lack of an adblocker or similar addons for the chosen browsing enviroment would be suggested also, third party javascript libraries are likely to be used and cannot be assured to function under such enviroments including modification or blocking of traffic/assets.

6.6 Development Requirements [18]

A number of softwares will be required by the development team in order to complete the project, these softwares may include:

- SQL management studio to manage the database. Azure is using TSQL which can be managed by SQL management studio on the developers local machine.
- Visual studio to develop the front and back-ends of the web application.
- Visual Studio Code to develop elements of the front-end
- Arduino IDE software for editing/flashing the boards in C/C++.

7 Limitations

It is important to acknowledge the limitations of this project, given that this project is taking place in a compressed format it will primarily be an exploratory exercise in IOT sensor technology and its ability to interface with middleware, frontend/backend and an SQL server implementation. In order to managing

expectations we are primary focused on heat,humidity and motion, this proof of concept implementation reduces the risk of under delivering on expectations.

7.1 Device Limitations

The ESP8266 is limited to the 2.4gHz wireless spectrum, and therefore a number of modern wireless networks may prove incompatible with the device. This is only a small hinderence to the device as most modern networks will use both 2.4 and 5.1gHz spectrums. Furthermore the 2.4gHz spectrum is better suited to longer distance connections than the 5.1gHz range, and even in the most heavily congested of wireless areas it would be reasonable to expect the payload from the ESP would have no issues at a measly 200bytes on average.

7.1.1 Security Concerns

Issues related to security on the publicly open API endpoints hosted in Azure prove to be hindered by the limitations of the devices. We are currently investigating potential fixes for the current implementation in which requests over https connections should ensure a level of security over current use of plain http.
Implementation of a method that would allow access to a preflashed public key to be used for encryption purposes of sensitive json data before it even leaves the device would be an excellent addition to the project, however both scope and inefficiencies of pub/private key encryption are reasons against this idea.

7.2 Development Limitations

Limitations of development experience may exist within the group with only 2 software engineering majors present. We expect that these members will be able to assist the other members in potential problem areas and expect the use of code versioning will alleviate any potential miscommunications in code modification and submission.
Factors potentially negating this issue include a base level of programming knowledge being shared across all members of the team, with a heavy level of experience in C# programming.

7.2.1 Framework Limitations

We may experience some level of difficulties with the chosen framework of .NETCore 2.0 in development as only a limited number of the team have experienced .NETCore MVC application programming or .NETCore API programming.
We expect however the time required for upskilling and learning the framework will provide a positive time return if not purely allowing team members a new skill in which to take to potential employment opportunities into the future.

References

[1] Forbes article: Top 8 IoT trends of 2018:
https://www.forbes.com/sites/danielnewman/2017/12/19/the-top-8-iot-trends-for-2018/

[2] Mckinsey article: Value of digitizing the physical world:
https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world

[3] Sun News article: China; Largest Renewables Producer:
http://sunnewsonline.com/china-is-worlds-largest-renewable-energy-producer-consumer/

[4] ESP8266 Information Page: http://esp8266.net/

[5] Jaycar XC-4444 Datasheet: https://www.jaycar.com.au/medias/sys_master/images/9105858396190/XC4444-dataSheetMain.pdf

[6] Jaycar Resources; DHT11 Datasheet: https://www.jaycar.com.au/medias/sys_master/images/9091897786398/XC4520-dataSheetMain.zip

[7] .NETCore SDK Download Page: https://www.microsoft.com/net/download

[8] TSQL Server Download Page:
https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-2017

[9] Azure Uptime SLA: https://azure.microsoft.com/en-au/support/legal/sla/app-service/v1_4/

[10] Arduino UNO3 Store Page: https://store.arduino.cc/usa/arduino-uno-rev3/

[11] Firefox: https://www.mozilla.org/en-US/firefox/

[12] Chromium: https://www.chromium.org/

[13] RaspberryPi Organisation Page: https://www.raspberrypi.org/

[14] OrangePi Website: http://www.orangepi.org/

[15] Spectre and Meltdown explaination page: https://meltdownattack.com/

[16] Unsplash Website: https://unsplash.com

[17] https://api.jquery.com/jQuery.ajax/

[18] Development resources:

Software/Hardware/Equipment	Description/Purpose
Visual Studio	Backend development
Visual Studio Code	Embedded Programming / Frontend development / Latex Typesetting
SQL Server 2016 Management Studio	Design and implementation of database
Microsoft Project	Gantt Chart Generation
GIMP	General Image Manipulation
Arduino IDE	Embedded programming
Azure Virtual Machine	Web Application Live Deployment
Slack	Team collaboration
GitHub	Code Repository

All image backgrounds are images from Unsplash, a provider of free and openly licensed images [16]