INFT3970 Major Project Scope Document
Distributed Monitoring System using Embedded Devices

Jay Rovacsek c3146220@uon.edu.au
Dean Morton c3252227@uon.edu.au

August 6, 2018

1 Project Scope Document

1.1 Introduction

The basic concept of this project will be to create a distributed system in which small devices are used to monitor, log and warn of select metrics from a multitude of potential data points.

1.2 Target Audience

The proposed target audience for this project are a wide range of individuals whom may wish to monitor any of the proposed metrics from the wireless devices. The design of this system allows for highly customisable systems that would be easy to imlement and preconfigured, with a view to write a management software that would allow end users to reconfigure the system as required at any point.

1.3 Metrics

Proposed metrics to be monitored include:

- Temperature
- Pressure (atmospheric)
- Humidity
- Sound (Decibels)
- Sound (Ultrasonic or Subsonic levels)
- Wind Speed
- UV
- Infared (Movement detection)
- Visible Light Levels
- Soil Moisture
- Acceleration
- Orientation/Tilt (x, y, z) of devices
- Rain Sensor
- Vibration
- CO^2 Sensors

1.4 Project Infastructure

Proposed hardware for the project would run on a mixture of ESP8266 or RaspberryPi like boards, made highly available of recent for reasonable prices.

Back-end hardware is proposed as any system that can host a number of services related to the project. The most essential of these services would include the storage of logging achieved by a database. Given the nature of open-source databases we can confidently assume either cloud hosting or localised hosting of a system is a non-concern.

1.5 Project Services

1.5.1 Database Service

The database service would be a simple API callable via a http post request, it would sit behind and MVC style application, allowing us to setup unique routes to define request types, or it could be a post of some json data which would define the table to be inserted into.

The later of these suggestions are less desirable as the devices could be reflashed to act negatively against the service host which likely would be centralised and not self-hosted in the deployed product.

1.5.2 User Management Service

A user management serice would be required to ensure security of an individuals privacy. This, as above would be best preflashed and not apparent to the end-user, taking the form of some kind of expirable token. The service would require some kind of method to associate a user with devices at the point of sale or some kind of method akin to that implemented by the Particle board range [1] which associates a serial or UUID/GUID code to identify the device in question. This would be pre-programmed at build time to be stored in the database so no management other than the entry of such a UUID/GUID would be required by the end-user.

1.5.3 Logging Service

The logging service would need to act as middleware between the devices and the database and potentially require some level of parsing of data and verification of data. Failing either of the parsing or authentication, the Logging service should push errors to the Error Management service.

Without errors occuring, the logging service should pass data required to the database service which would perform required tasks to store such data.

1.5.4 Error Handling Service

The Error handling service would log errors from either data pushed by end devices or the websystem itself, the logs would include stack-trace material, cause for logging, associated user data and timestamps.

1.5.5 Web Service

The web service should include all elements required to report a user-account's associated data for selected periods. This should also allow the update of an account, or deletion of an account. Included requiremets of this service will include but are not limited to:

- Adding or Removing a device from an account
- Visualisation of data associated with an account
- Login / Logout functionailty
- Downloadable payload of data associated with an account in various formats

1.6 Arguements for this project

A number of arguements for this project include but are not limited to:

- Numerous potential metrics to measure that could prove to be interesting
- Interesting field for multiple team members
- Varied required services that would allow allocation of tasks to prove easy

1.7 Arguements against this project

A number of arguements against this project include but are not limited to:

- Steep learning curve for low level device programming
- Multiple required services to interface with such basic devices
- Requirements of the real world
  - Do people need / want / require this?
  - Could people implement this easily themselves?

References

[1] https://www.particle.io/