



IoT Using LoRaWAN
CSE3PR Industry Project
Team F – System
Maintenance Document

James Curnow

Joel Morran

Michael Owczarek

Remi Petit

Luis Shigetomi

Copyright © June 2019

Table of Contents

1.0 – Introduction

- 1.1 – System Overview
- 1.2 – Audience Description
- 1.3 – Applicability Statement
- 1.4 – Purpose Statement
- 1.5 – Document Usage Description
- 1.6 – Conventions
- 1.7 – Change Log

2.0 – Software Design Scope

- 2.1 – Major Software Functions
- 2.2 – Major Design Constraints and other Requirements

3.0 – Reference Documents

- 3.1 – Existing Software Documentation
- 3.2 – System Documentation
- 3.3 – Vendor Documentation
- 3.4 – Other Documentation

4.0 – User Stories

- 4.1 – User Story Dictionary
- 4.2 – Iterative User Story Documents
 - 4.2.1 – User Story Definition
 - 4.2.2 – Flow of Interaction Diagram
 - 4.2.3 – User Story Testing
 - 4.2.4 – Wireframes

5.0 – Object-Oriented Design

- 5.1 – High Level System Architecture
- 5.2 – Use Case Analysis
- 5.3 – Database/ERD

6.0 – Software Release Report

6.1 – Usability Test Report

6.2 – System (User Story) Test Report

7.0 - Setup/Installation Guide

7.1 – Raspbian Installation

7.2 – Gateway (Raspberry Pi) Setup

7.2.1 – Installing the Container Runtime

7.2.2 – Installing the IoT Edge Security Daemon

7.2.3 – Connecting to your IoT Hub

7.2.4 – Verifying Successful Installation

7.2.5 – Enabling Interfacing Options

7.3 – Arduino Setup

7.4 – Azure Setup

7.5 – Power BI Setup

8.0 – Appendix

8.1 – Glossary

8.2 – Statement of Effort

8.3 – Statement of Authorship

Table of Diagrams

Figure 1 – Block diagram – Page 5

Figure 2 – Flow of Interaction diagram – Page 12

Figure 3 – Wireframe – Page 14

Figure 4 – System Architecture Diagram – Page 15

Figure 5 – ERD – Page 16

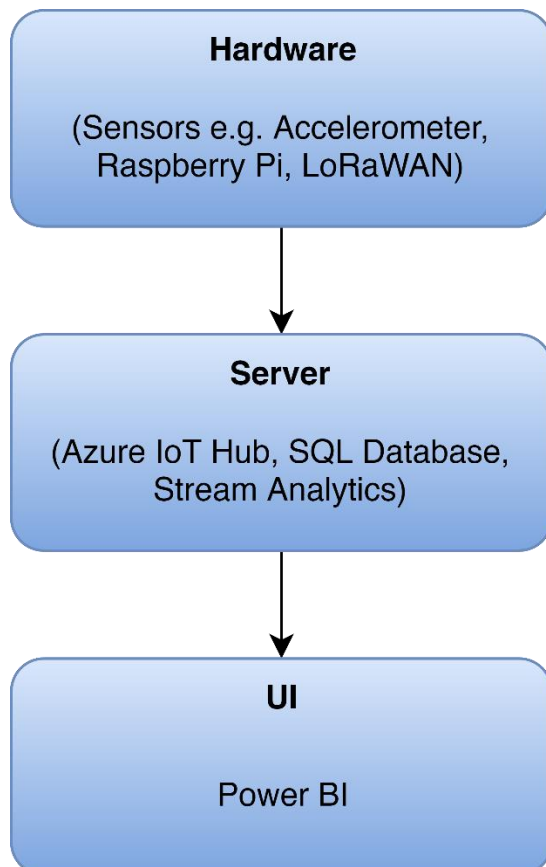
1.0 – Introduction

1.1 – System Overview

The IoT Using LoRaWAN system aims to deliver the following outcomes:

- A group of sensors connected through LoRaWAN to a gateway
- A connection through the gateway directly to the IoT platform for sending and receiving the collected data
- An IoT platform set up for managing the data to and from the sensors
- A visualisation environment to illustrate the sensors activities
- A data processing algorithm to extract features and detect certain patterns in the logged data

The following block diagram portrays the system:



Hardware: Consists of the physical components of the software responsible for sensor data transmission to the server side of the project. This consists of a Raspberry Pi, a Seeeduno with GPS and a Grove Accelerometer.

Server: Microsoft's Azure platform will be used for anything server side. This includes an IoT Hub used to manage the IoT devices as well as the system's database.

User Interface: The UI components are provided by Microsoft's Power BI.

1.2 – Audience Description

There are two types of users for the system:

Primary: The IoT using LoRaWAN project is intended for use by staff at the LaTrobe University Centre for Technology Infusion.

Secondary: The system is also being developed with the hope that it can be marketed and used by various people with specific needs. For example, farmers could use the system in order to track their cattle's behavior and movement patterns

For both user types, basic computer skills are required. Knowledge of how Power BI works is beneficial but not required.

1.3 – Applicability Statement

In terms of hardware, the system runs on Raspberry Pi and Arduino sensors.

In terms of software, the system is managed entirely through Microsoft's Azure.

The system is accessed by the user using any web browser that supports Microsoft's Power BI. The user can also access the Azure IoT Hub online should they ever want to manage their devices.

1.4– Purpose Statement

The purpose of the IoT using LoRaWAN project is to:

- Create a visualization environment for sensor activity
- Provide an environment capable of handling additional sensors
- Provide separate restricted views of data for users based on their role in the system
- Display sensor specific data in an understandable format

This project's aim is to implement a LoRaWAN sensor network with a few sensors to generate data and present/visualise it for a user.

1.5 – Document Usage Description

1 – Introduction:

Consists of a short description outlining an overview of the system, it also discusses the audience and provides purpose of the usage of the document. The conventions used on the document and changelog are also included.

2 – Software Design Scope:

Outlines a couple of the software components that are used in a predominately hardware-based project. Any design constraints and further requirements are also included.

3 – Referencing Documents:

Provides a detailed description of any documentation used in relation to the system.

4 – User Stories:

Consists of user stories that are taken into the consideration of the system. A user story dictionary as well as testing and wireframes are also included.

5 – Object-Oriented Design:

Any diagrams and descriptions relating to the architecture of the system are included in this part of the document.

6 – Software Release Report:

System usability and any testing performed are included in this section of the documentation.

7 – Setup/Installation Guide:

Describes how to setup up the system and install the necessary hardware.

8 – Special Notes:

Additional information is included in this section of the document such as a glossary or any further comments that are not included elsewhere.

1.6 – Conventions

The diagrams in this document all follow the standards of UML.

1.7 – Change Log

Date	Change Made	Changed By
11 June 2019	Final changes made	Michael Owczarek, James Curnow, Joel Morran
10 June 2019	Further revised formatting	All
9 June 2019	Revised formatting	All
7 June 2019	Revised document created	All
7 June 2019	Changelog	James Curnow
3 June 2019	Iterative user story documents	Michael Owczarek
29 May 2019	Table of contents	Michael Owczarek
29 May 2019	Gateway setup	Joel Morran
26 May 2019	Power BI installation	Luis Shigetomi
26 May 2019	User stories	Joel Morran
26 May 2019	Documentation	Remi Petit
25 May 2019	Document usage description	James Curnow

2.0 – System Design Scope

2.1 – Major System Functions

The main function of the system is to record and display various behavioral and movement patterns of a subject (person or animal). This is achieved by using an accelerometer and GPS attached to the subject which is then sent to Power BI via a LoRaWAN gateway and then displayed to the user using various analytical tools.

User:

- View captured data in the form of graphs and charts
- Export the data in various formats

System:

- Capture data from an Arduino GPS and accelerometer
- Send data from Arduino to LoRaWAN gateway
- Decode the data and send it to an Azure database
- Performs various analytics on the data in the database

2.2 – Major Design Constraints and Other Requirements

Constraints:

The cost of running the Azure deployment must be considered as running a stream analytics job can be very cost intensive.

The project inherits any constraints from the systems and components (Azure, Power BI, Raspbian, etc.) that it utilises.

Other Requirements:

The LoRaWAN gateway requires a strong, stable network in order to function properly and send each packet to Azure.

Power BI also requires a persistent internet connection.

3.0 - Reference Documents

3.1 – Existing Software Documentation

The following documents are supplied in the final submission folder:

- 2019 Industry Project Summary.pdf
- LoraWan CTI setup.pdf

3.2 – System Documentation

The software is not being developed from an embedded existing system.

3.3 – Vendor Documentation

Microsoft Azure: Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers.

<https://azure.microsoft.com/en-au/>

<https://docs.microsoft.com/en-us/azure/iot-edge/how-to-install-iot-edge-linux-arm>

<https://azure.microsoft.com/en-au/resources/samples/custom-vision-service-iot-edge-raspberry-pi/>

Raspberry Pi: The Raspberry Pi is a series of small single-board computers used to promote teaching of basic computer science in schools and in developing countries.

<https://www.raspberrypi.org/documentation/>

Arduino: Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits.

<https://www.arduino.cc/en/Main/Docs>

Seeeduino: Seeeduino V4.2 is an Arduino-compatible board, which is based on ATmega328P MCU.

http://wiki.seeedstudio.com/LoRa_LoRaWan_Gateway_Kit/

http://wiki.seeedstudio.com/Seeeduino_LoRAWAN/

http://wiki.seeedstudio.com/Grove-IMU_9DOF_v2.0/

Power BI: Power BI is a business analytics service by Microsoft. It aims to provide interactive visualizations and business intelligence capabilities with an interface simple enough for end users to create their own reports and dashboards.

<https://powerbi.microsoft.com/en-us/learning/>

<https://docs.microsoft.com/en-us/power-bi/desktop-getting-started>

Python: Python is an interpreted, high-level, general-purpose programming language.

<https://docs.python.org/3/>

3.4 – Other Documentation

The following documents are supplied in the final submission folder:

- References.docx

4.0 – User Stories

4.1 – User Story Dictionary

“As a user, I want a clean UI, so that I can easily understand and interpret the behavioural patterns of a given subject”

“As a user, I want the software and hardware to be easy to setup, so that I can quickly and easily manage my devices”

“As a user, I want the solution to be cost effective, so that I can use the system and not worrying about the cost”

“As a user, I want the data to be easy to export, so that I can interpret the data on another platform if needed/wanted”

4.2 – Iterative User Story Documents

4.2.1 – User Story Definition

Story ID: US-1

Story: “As a user, I want a clean UI, so that I can easily understand and interpret the behavioural patterns of a given subject”

Story Description: The Power BI interface needs to be clear, concise and simple to understand for the user.

Story ID: US-2

Story: “As a user, I want the software and hardware to be easy to setup, so that I can quickly and easily manage my devices”

Story Description: The system needs to be simple to setup so that multiple sensors can quickly be deployed. The setup also needs to be easy to setup so that even people with an average level of computer skills can do it. It also needs to be easy for devices to be added and removed.

Story ID: US-3

Story: “As a user, I want the solution to be cost effective, so that I can use the system and not worrying about the cost”

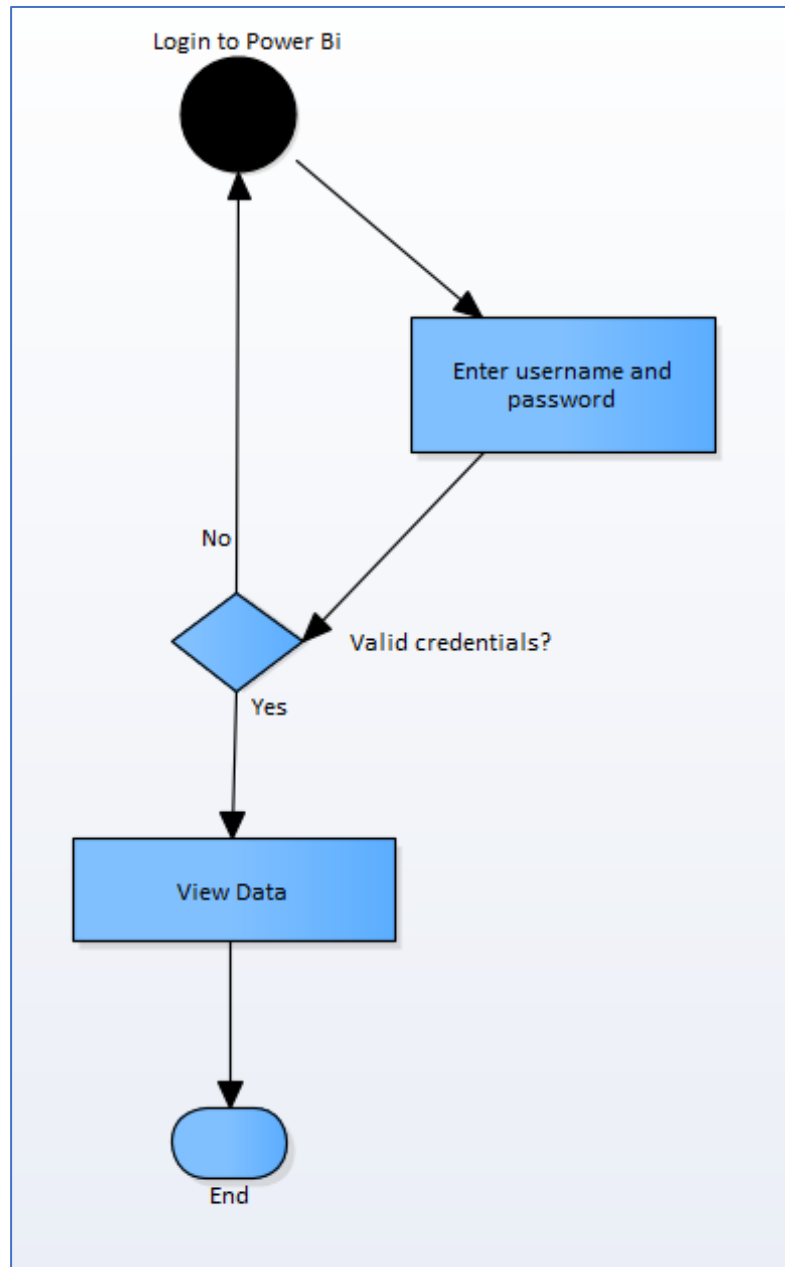
Story Description: The system needs to be fine-tuned so that it does not burn through any budgeting in place. Keeping Azure running 24/7 is the main cost for the system.

Story ID: US-4

Story: “As a user, I want the data to be easy to export, so that I can interpret the data on another platform if needed/wanted”

Story Description: A User might want to look at the collected data on a different platform instead of Power BI and will want a simple way of exporting it.

4.2.2 – Flow of Interaction Diagram

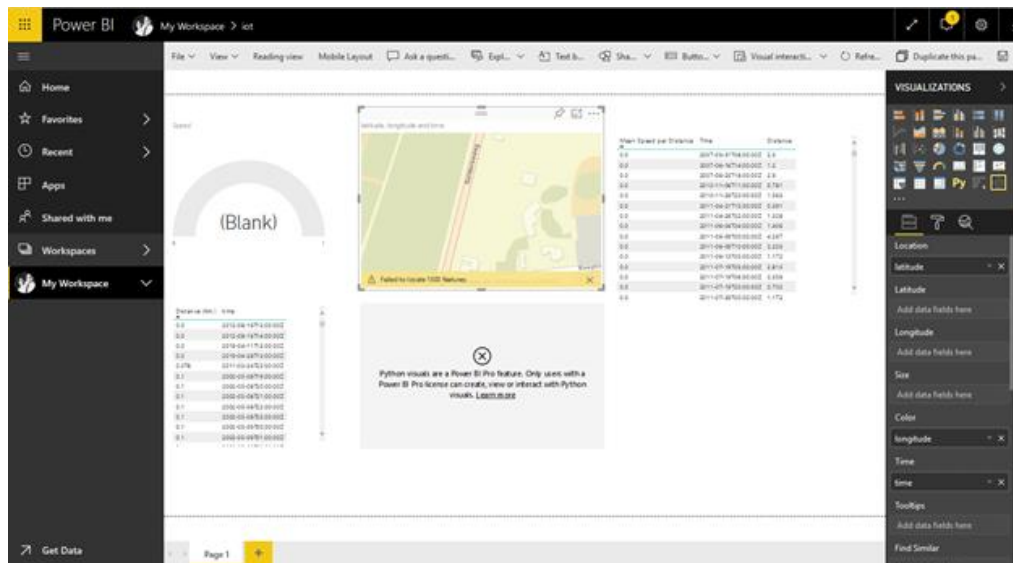


4.2.3 – User Story Testing

User Story	Test ID	Acceptance Criteria	Description	Status
US-1	US1T-1	Easy to understand and Clean UI	Users do not have difficulty understanding and navigating the UI	PASS
	US1T-2	Hard to understand and Clunky UI	Users have difficulty understanding and navigating the UI and find it confusing and clunky	FAIL
US-2	US2T-1	Easy setup	Users find the setup process to be easy based on the documentation provided	PASS
	US2T-2	Hard/complex setup	Users find the setup process to be hard and complex and to technical for them based on their skill level and poor documentation/no documentation provided	FAIL
	US2T-3	Setup requires and expert	The setup requires and expert to complete the process	FAIL
US-3	US3T-1	Cost effective	Cost are within/ below extreme low-cost mass production targets	PASS
	US3T-2	Not cost effective	Cost exceed extreme low-cost mass production targets	FAIL
US-4	US4T-1	Data easy to export	Data is easy to export to desired data format should support wide range of formats	PASS
	US4T-2	Hard to export	Data is not easy to export and only supports limited formats	FAIL
	US4T-3	No export supported	Data cannot be exported	FAIL

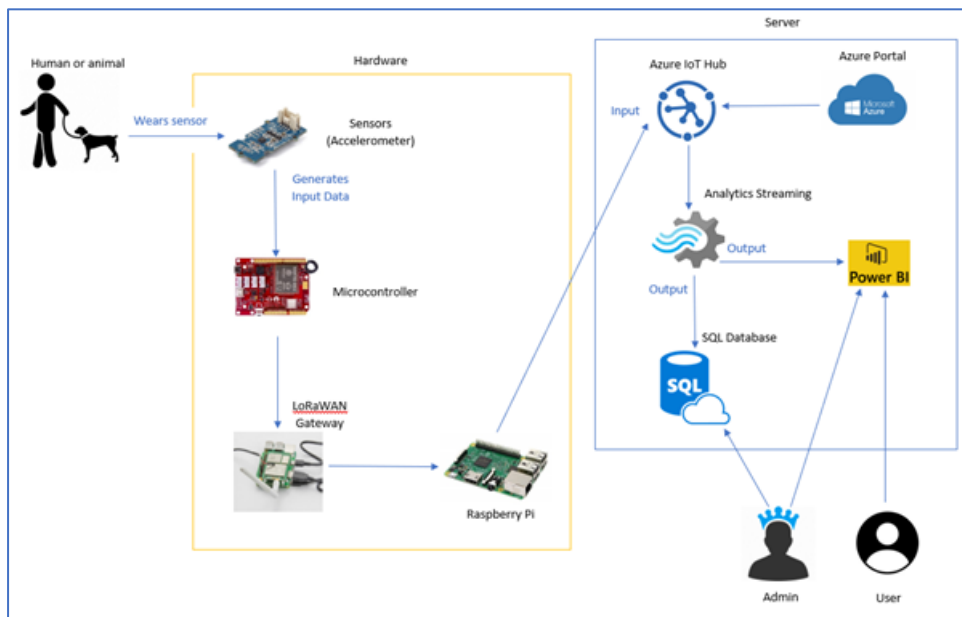
4.2.4 – Wireframes

The only interface designed for this system is a single Power BI report:

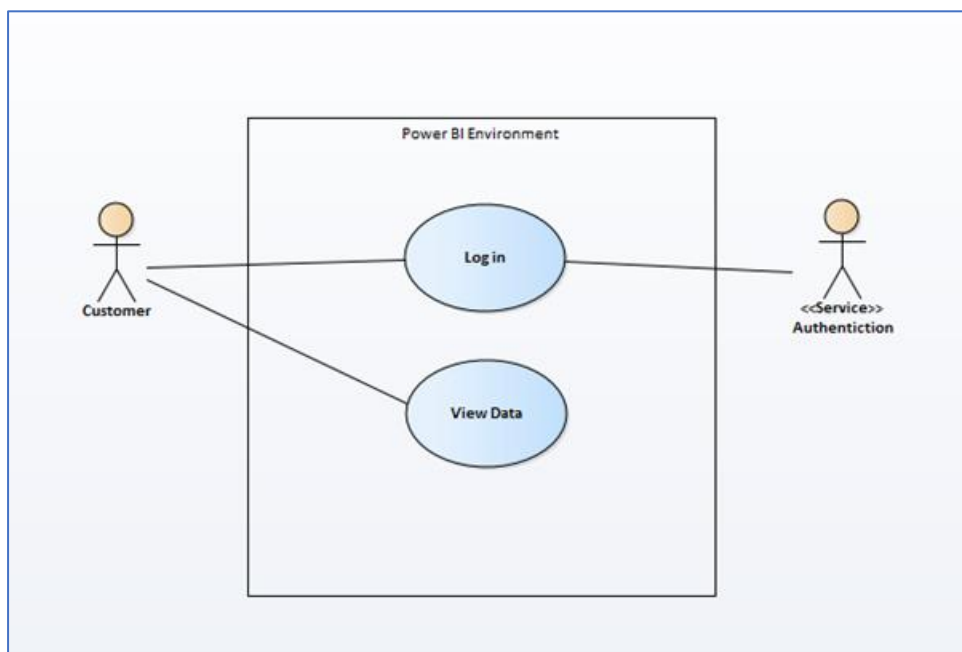


5.0 – Object-Oriented Design

5.1 – High Level System Architecture



5.2 – Use Case Analysis



USE CASE 1: User Login

TRIGGER/GOAL: A user wants to login to Power BI

ACTOR: Admin/User

MAIN FLOW

1. User enters username into the username field and password into the password field
2. User is directed to Power BI dashboard

EXTENSIONS

- 1a - User enters the wrong credentials. Step 1 in main flow is repeated

USE CASE 2: User view data

TRIGGER/GOAL: A user wants to view the data

ACTOR: Admin/User

MAIN FLOW

1. The User opens their chosen view from Power BI dashboard
2. User is directed to a Power BI report where they can see the requested behavioural and movement patterns

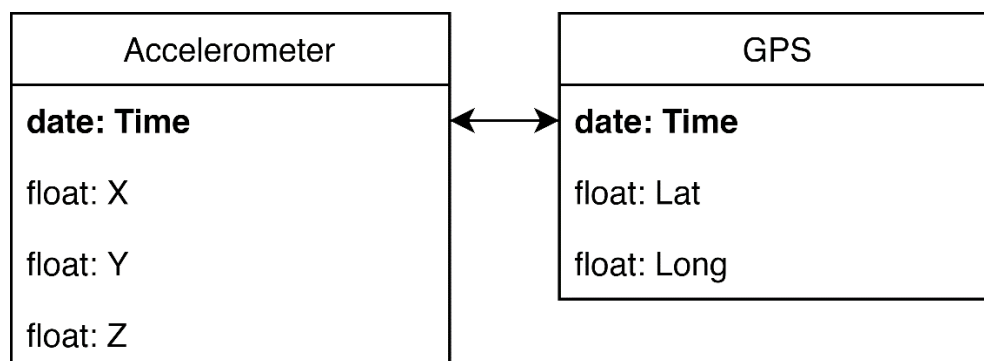
EXTENSIONS

- 1a - User enters the wrong credentials. Step 1 in main flow is repeated.

5.3 – Database/ERD

The system uses a simple Azure SQL database.

The ERD is as follows:



6.0 – Software release report

6.1 – Usability Test Report

For usability, we will test our application with different people from different age groups. We will ask them to look at our live reports and give their review and rating of how the complex the data seems to them and if it is easy for them to understand or not.

Since we have not implemented the system fully yet and don't have much working sample data, we have not performed many tests until now.

The test we perform is simple as we are only showing a single page that shows the report on it and example is listed below:

Name: John Smith

Age: 27

Gender: Male

Computer Skills: Average

Overall Rating: 7.5/10

Functionality	Comments	Rating (Out of 10)
Login	xxx	7
Data easy to interpret	xxx	3
Data easy to export	xxx	9
Live data updates at a relevant pace to the user	xxx	5
Overall report design	xxx	4

6.2 – System (User Story) Test Report

Test Id	Test Purpose	Steps	Input Parameters	Actual Data Input	Expected Output	Test Status
1.0	User Login	1.Enter login details 2.Click login	"Username" "Password"	"Username" "Password"	The user should be logged into Power Bi	Success
2.0	View Data	1.View the dashboard	N/A	N/A	The user should be able to view the graphical data on the dashboard	Success

7.0 - Setup/Installation Guide

7.1 – Raspbian Installation

1. Download Raspbian.

The following image is tested and working for a Raspberry pi 3 B+

<https://downloads.raspberrypi.org/raspbian/images/raspbian-2018-11-15/>

2018-11-13-raspbian-stretch.zip, 2018-11-13 14:13, 1.0G

2. Install Raspbian to the microSD card.

Download and install the Etcher SD card burner utility.

<https://www.balena.io/etcher/>

3. Run Etcher and select the Raspbian image that you extracted in step 1.

Select the microSD card drive. Etcher may have already selected the correct drive.

Click Flash to install Raspbian to the microSD card.

Remove the microSD card from your computer when installation is complete.

4. Insert the microSD card into Pi.

7.2 – Gateway (Raspberry Pi) Setup

7.2.1 – Installing the Container Runtime

The following commands install both the Moby-based engine and command-line interface (CLI). The CLI is useful for development but optional for production deployments.

Paste and run the follow commands (without the single quotes) one at a time into your Raspberry Pi terminal:

```
'curl -L https://aka.ms/moby-engine-armhf-latest -o moby_engine.deb && sudo  
dpkg -i ./moby_engine.deb'
```

```
'curl -L https://aka.ms/moby-cli-armhf-latest -o moby_cli.deb && sudo dpkg -i  
./moby_cli.deb'
```

```
'sudo apt-get install -f'
```

7.2.2 – Installing the IoT Edge Security Daemon

Paste and run the follow commands (without the single quotes) one at a time into your Raspberry Pi terminal:

```
'curl -L https://aka.ms/libiothsm-std-linux-armhf-latest -o libiothsm-std.deb &&  
sudo dpkg -i ./libiothsm-std.deb'
```

```
'curl -L https://aka.ms/iotedged-linux-armhf-latest -o iotedge.deb && sudo dpkg -i  
./iotedge.deb'
```

```
'sudo apt-get install -f'
```

7.2.3 – Connecting to your IoT Hub

To manually provision a device, you need to provide it with a device connection string that you can create by registering a new IoT Edge device in your IoT hub.

1. Open the configuration file in terminal using the following command:

```
'sudo nano /etc/iotedge/config.yaml'
```

2. The text document should look as follows:

```
yaml  
  
provisioning:  
  source: "manual"  
  
  device_connection_string: "<ADD DEVICE CONNECTION STRING HERE>"  
  
# provisioning:  
#   source: "dps"  
#   global_endpoint: "https://global.azure-devices-provisioning.net"  
#   scope_id: "{scope_id}"  
#       registration_id: "{registration_id}"
```

3. Save and close the file.

```
'CTRL + X', 'Y', 'Enter'
```

4. After entering the provisioning information in the configuration file, restart the daemon:

```
'sudo systemctl restart iotedge'
```

7.2.4 – Verifying Successful Installation

You can check the status of the IoT Edge Daemon by entering the following commands in terminal:

'systemctl status iotedge'

'journalctl -u iotedge --no-pager --no-full'

'sudo iotedge list'

7.2.5 – Enabling Interfacing Options

Enable VNC, SPI, Serial, SSH and I2C

1. Click the **'Raspberry icon'** > **'Preferences'** > **'Raspberry Pi Configuration'**.
2. On the Interfaces tab, set **'VNC'**, **'SPI'**, **'Serial'**, **'SSH'** and **'I2C'** to Enabled, and then click **'OK'**.
3. Reboot the Raspberry Pi

7.3 – Arduino Setup

1. Download & install the latest version of Arduino IDE
2. Open Arduino IDE, select **'File'** > **'Preferences'**
3. Paste **'https://raw.githubusercontent.com/Seeed-Studio/Seeed_Platform/master/package_seeeduino_boards_index.json'** into **'Additional Boards Manger URLs'**
4. Select **'Tools'** > **'Board'** > **'Board Manager'**
5. Search for and install **'Seeed SAMD Boards'**
6. Select **'Tools'** > **'Board'** > **'Seeeduino LoRaWAN'**
7. If not done already, connect the Arduino board to the PC.

Note: The driver for the Seeeduino should automatically install. If it doesn't use the driver provided here:

https://github.com/SeeedDocument/Seeeduino_LoRa/raw/master/res/driver.zip

8. Select **'Tools'** > **'Port'** and select the Seeeduino COM port.
9. Select **'Open'** and select **'LoRaWAN.ino'**
10. Select **'Upload'** and wait for the code to be uploaded onto the Seeeduino.

Note: Uploading to the Seeeduino is known to be temperamental, so if it doesn't work on the first try, just try again.

11. The Seeeduino is now setup and will function whenever powered via battery or USB

7.4 – Azure Setup

7.4.1 – Resource Group Setup

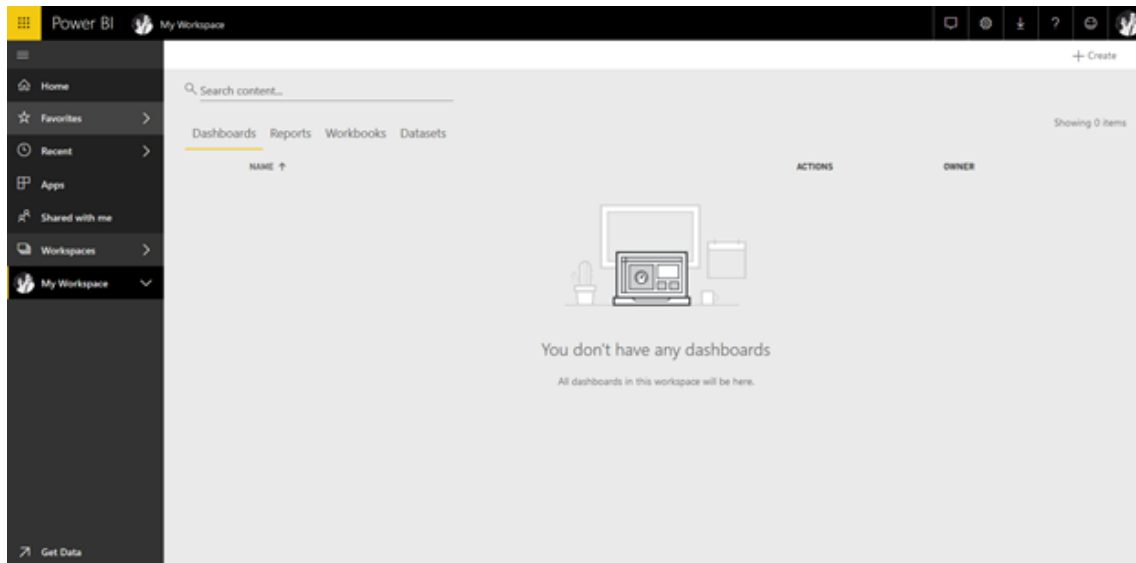
1. Navigate to <https://portal.azure.com/#create/Microsoft.ResourceGroup>
2. Enter in any name for **'Resource group'**
3. Set **'Region'** to a region of your choice (**'(Asia Pacific) Australia East'** is recommended)
4. Select **'Review + Create'**
5. After verifying the details, select **'Create'** and wait for Azure to confirm setup of the resource group

7.4.2 – Template Deployment

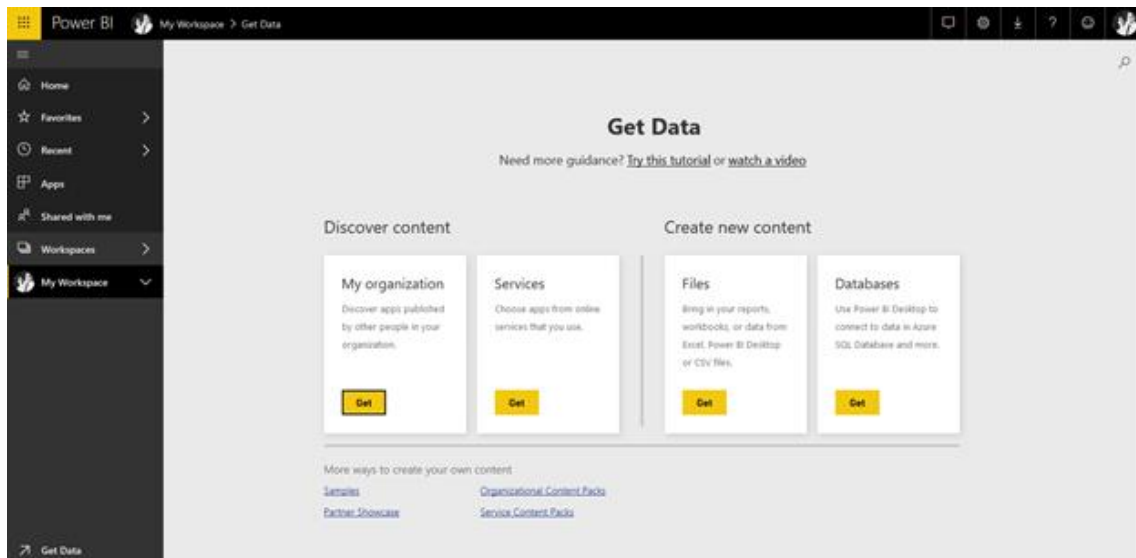
1. Navigate to <https://portal.azure.com/#create/Microsoft.Template>
2. Select **'Edit template'**
3. Select **'Load file'** and then select **'template.json'**
4. Select **'Save'**
5. Set **'Resource group'** to the same resource group that was created in
6. Select **'Purchase'** and wait for Azure to finish the deployment (which should take around 10 mins to complete)

7.5 – Power BI Setup

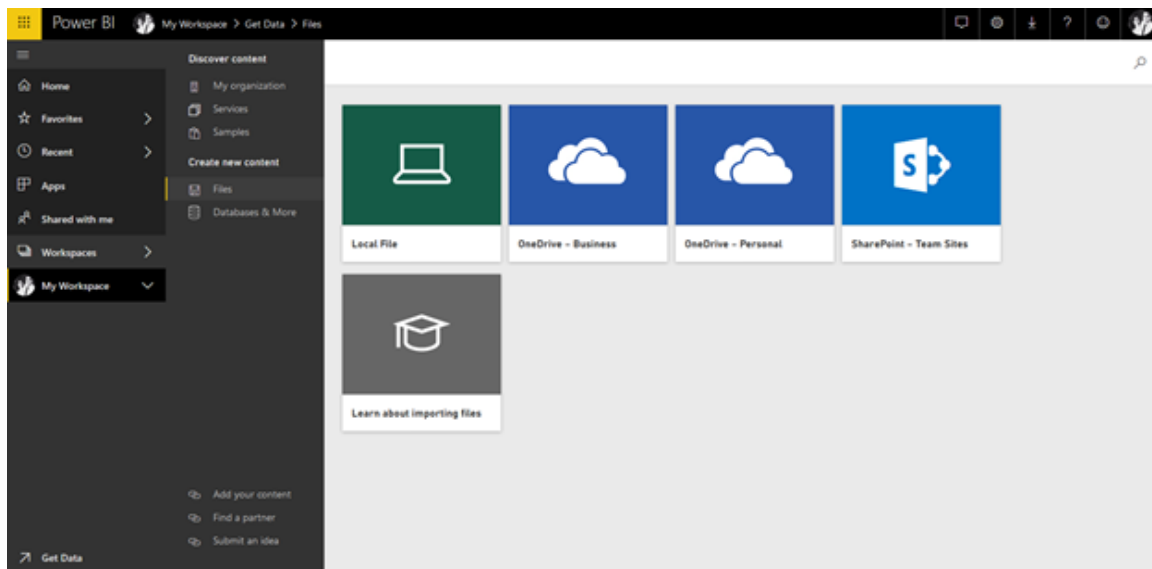
1. Navigate to <https://app.powerbi.com/home> and login with your account.
2. On the bottom left, select 'Get Data'



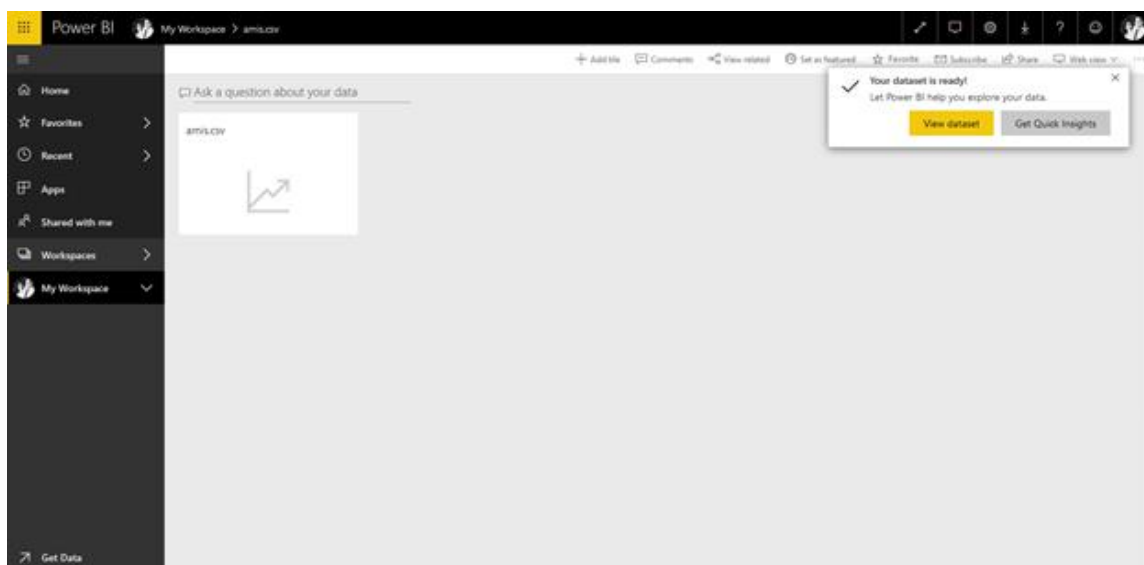
3. In Create new content, click on 'Get' under Files



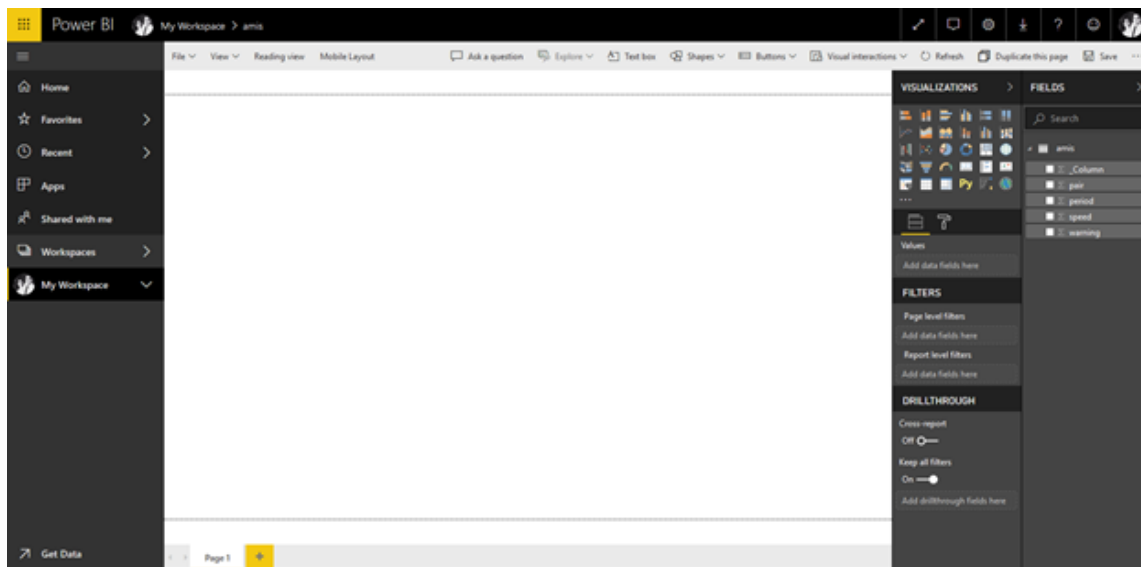
- This will take you to a screen where you can pick the file to upload, whether it is from the cloud or a local file.



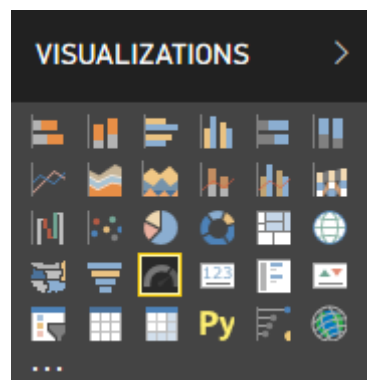
- Click on Local File and select your .csv or excel file to load the dataset to power bi

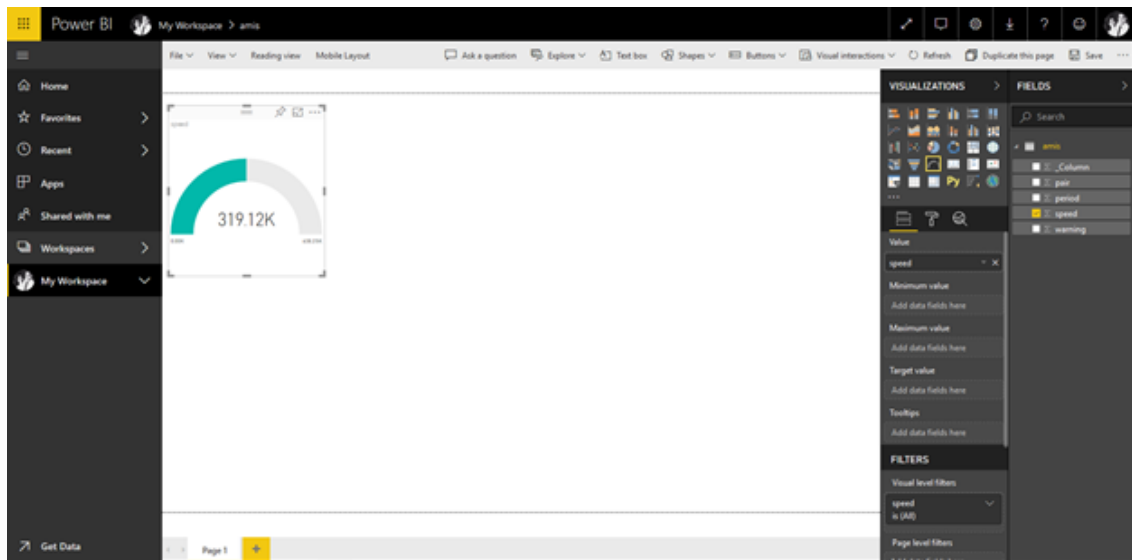


6. If loaded correctly, you should see on the right side of the screen under Fields section, the tables from the database.



7. Under Visualizations, select desired graph, and click on it once, it will transfer the graph to the working area. Once the graph is selected you can select the field or fields desired from your tables, and it will fill the graph with colors and numeric values.

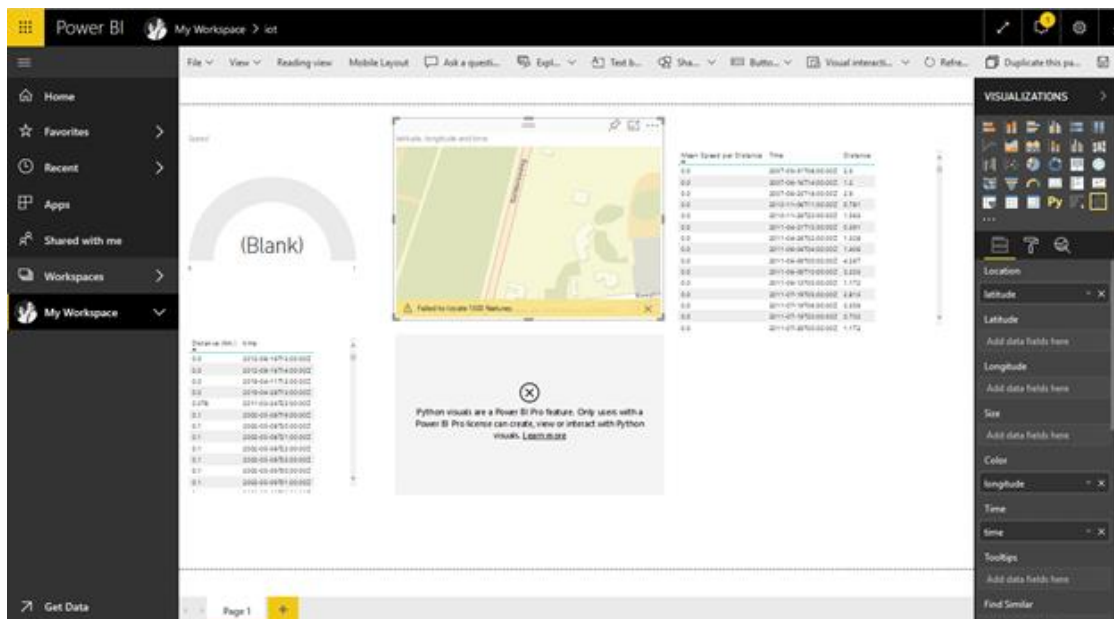




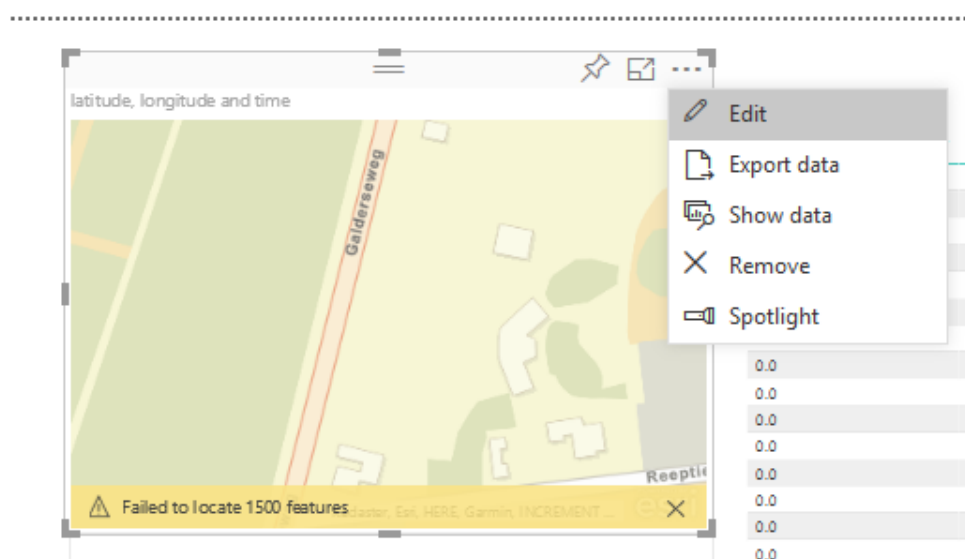
8. In the value section, you can modify or pick how you want the data to behave. Eg. Average, sum, etc. The same applies with the rest of the graphs.



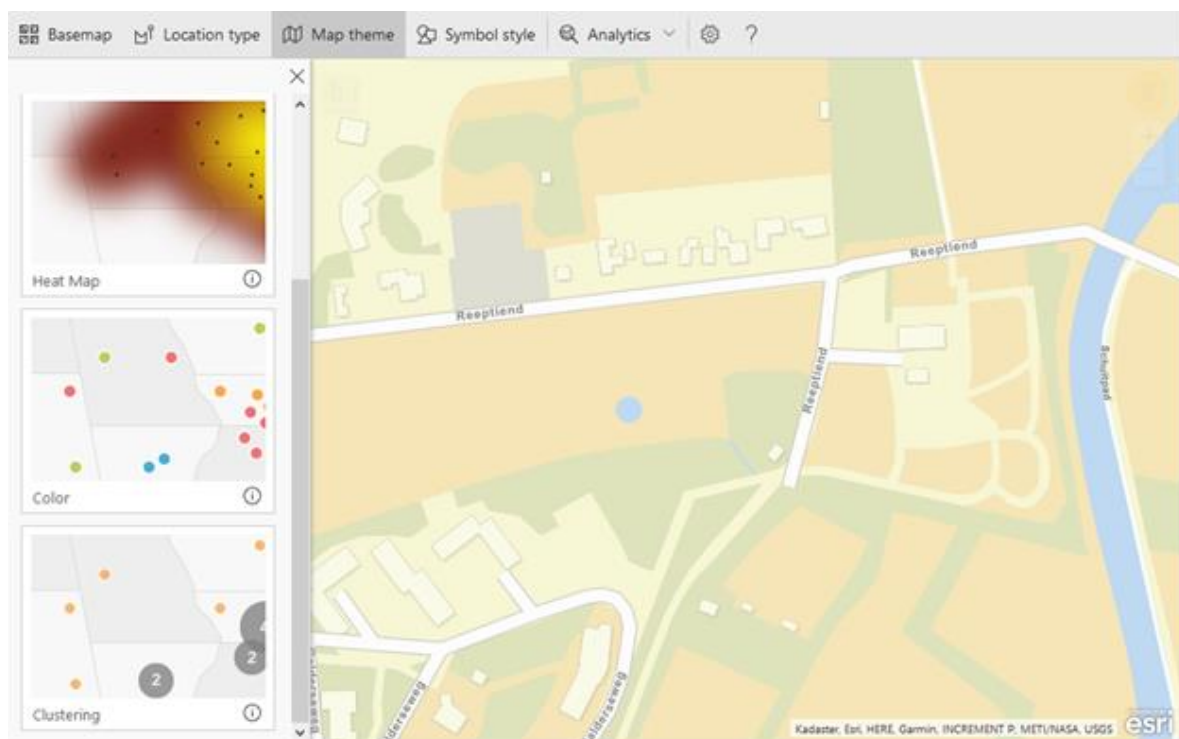
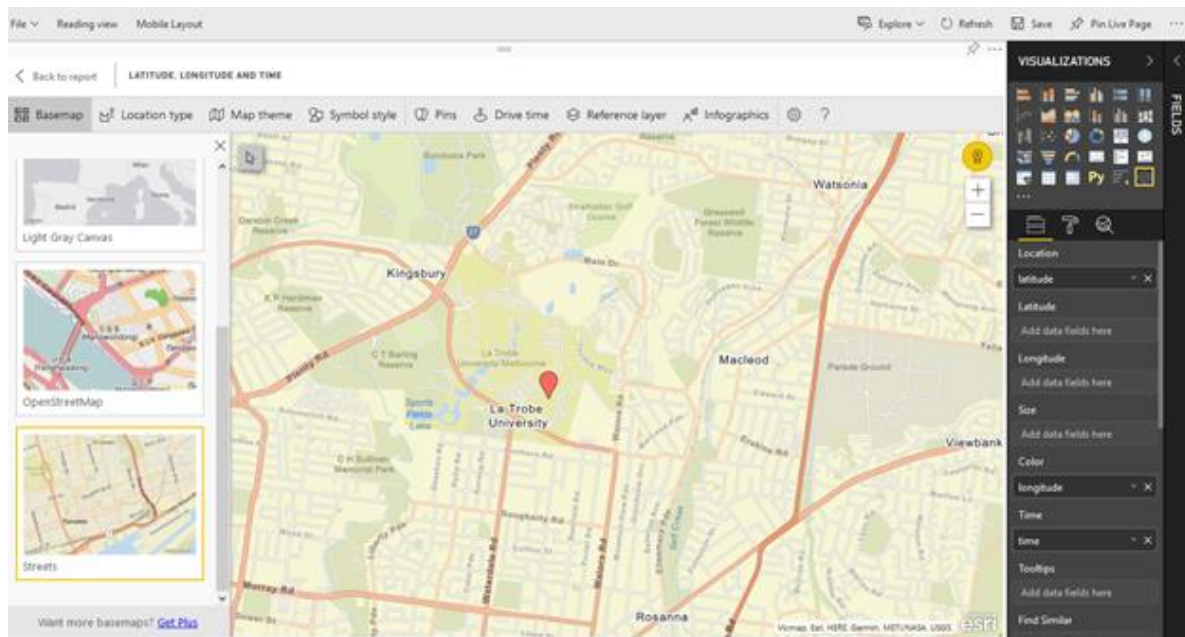
9. To add a map click on any globe icon in Visualizations and select latitude and longitude values

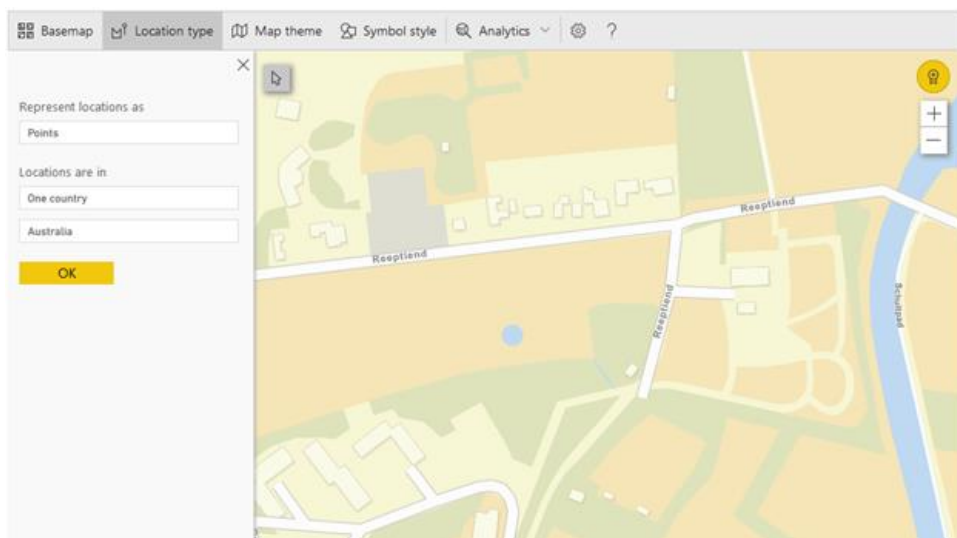


10. Maps are also editable, such as colors and themes. To do this, click on the ... on the upper right side of the map, and select edit

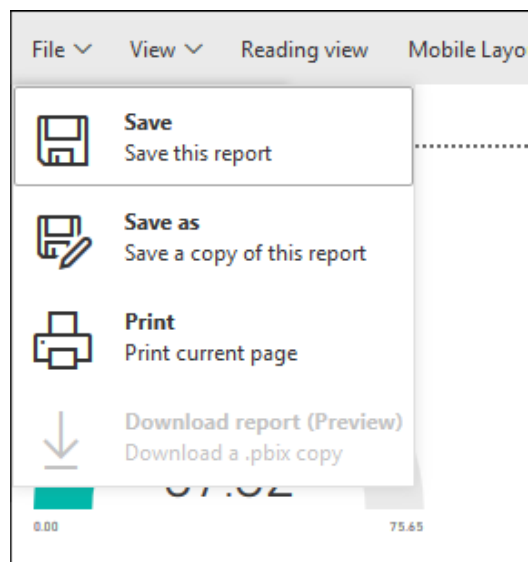


Feel free to edit as needed.





11. Once you have finished your work, save the report



Dashboards Reports Workbooks Datasets				Showing 2 items	
NAME ↑		ACTIONS		OWNER	
	☆ iot			Luis Steve Shigetomi	
	☆ test			Luis Steve Shigetomi	

8.0 – Appendix

8.1 – Glossary

IoT:

The Internet of things is the extension of Internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity, and other forms of hardware, these devices can communicate and interact with others over the Internet, and they can be remotely monitored and controlled.

LoRaWAN:

LoRaWAN is a media access control (MAC) protocol for wide area networks. It is designed to allow low-powered devices to communicate with Internet-connected applications over long range wireless connections.

SQL:

SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.

Microsoft Azure:

Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers.

Raspberry Pi:

The Raspberry Pi is a series of small single-board computers used to promote teaching of basic computer science in schools and in developing countries.

Arduino:

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits.

Seeeduino:

Seeeduino V4.2 is an Arduino-compatible board, which is based on ATmega328P MCU.

Power BI:

Power BI is a business analytics service by Microsoft. It aims to provide interactive visualizations and business intelligence capabilities with an interface simple enough for end users to create their own reports and dashboards.

Python:

Python is an interpreted, high-level, general-purpose programming language.

8.2 – Statement of Authorship

Department of Computer Science and Computer Engineering STATEMENT OF AUTHORSHIP FOR GROUP ASSIGNMENTS

We each certify that our own contribution in the attached material is original work and that we have no knowledge of an act of plagiarism committed by any member of this group. We declare that no other person's work has been used without due acknowledgement. Except where we have clearly stated that we have used some of this material elsewhere, it has not been presented by us for examination in any other course or unit at this or any other institution.

We have read the Academic Integrity Statute and completed the Academic Integrity Module relating to Student Discipline and Academic Misconduct, which are available on the University's website and understand that we are bound by such Policy, Statute and Regulation and that we may be subject to student discipline processes in the event of an act of plagiarism by us.

We understand the nature of plagiarism to include the reproduction of someone else's words, ideas or findings and presenting them as our own without proper acknowledgement. Further, we understand that there are many forms of plagiarism which include direct copying or paraphrasing from someone else's published work (either electronic or hard copy) without acknowledging the source; using facts, information and ideas derived from a source without acknowledgement; producing assignments (required to be independent) in collaboration with and/or using the work of other people; and assisting another person to commit an act of plagiarism. We understand that the work submitted may be reproduced and/or communicated by the University or a third party authorised by the University for the purpose of detecting plagiarism.

- Unit Code: **CSE3PRA**
- Unit Name: **INDUSTRY PROJECT A**
- Lecturer/Tutor's Name: **Dr Amin B. Abkenar**
- Supervisor's Name: **Sameer Muhammad**
- Name of Document: **System Maintenance Document**

Student No.	Family Name	Given Name	Signature	Date
18938149	Petit	Remi	Remi Petit	9/06/19
18937397	Owczarek	Michael	Michael Owczarek	9/06/19
17699381	Morran	Joel	Joel Morran	9/06/19
19331567	Curnow	James	James Curnow	9/06/19
19145764	Shigetomi	Luis	Luis Shigetomi	9/06/19

Plagiarism

One form of academic cheating is plagiarism, the reproducing of someone else's words, ideas or findings and presenting them as one's own without proper acknowledgment.

There are many forms of plagiarism, including the following:

1. Direct copying of sentences, paragraphs or other extracts from someone else's published work (including on the Internet and in software) without acknowledging the source;
2. Paraphrasing someone else's words without acknowledging the source;
3. Using facts and information derived from a source without acknowledging it;
4. Using ideas directly derived from an identifiable author without acknowledging the source;
5. Producing assignments which should be the student's own, independent work in collaboration with and/or using the work of other people (e.g. a student or tutor).

Assisting another person to plagiarise material may be punished as severely as is plagiarism itself. Assisting plagiarism may involve a student lending work (or by posting it on the Internet for sale) which is intended for submission for assessment, or which has already been submitted so that it can be copied and handed in by another student as that student's own work.³

Students may find it helpful to discuss assignments with other students and their tutors. The University encourages students to communicate with one another in constructive ways about the learning process. Students may choose to assist each other, for example in discussing the approaches that might be taken to assignment topics or helping with the availability of reading materials. They should, however, write their assignments independently, except when they are asked to work on a project as a member of a group which is to submit a joint report, as equal contributors. In this case, except where the final written work consists of sections for which particular individuals take sole responsibility, the group as a whole assumes responsibility for it and the proper acknowledgment of any use made of the words or ideas of people outside the group.

The increased vigilance of the University in regard to plagiarism is in response to the perceived growing incidence of plagiarism. The new recommended penalties reflect the value the University places on academic honesty and how seriously it regards plagiarism offenses. The University will protect its reputation for academic integrity by ensuring that serious and serial offenders, who wish to obtain an unfair advantage through cheating, are given penalties proportionate to the offenses committed.

8.3 – Statement of Effort

Section	Members
1.0 - Introduction	James
2.0 - System Design Scope	Michael
3.0 - Reference Documents	Remi
4.0 - User Stories	Joel, Michael, Luis
5.1 - High Level System Architecture	James
5.2 - Use Case Analysis	James
5.3 - Database/ERD	Michael
6.0 - Software Release Report	Joel
7.1 - Raspbian Installation	Joel
7.2 - Gateway (Raspberry Pi) Setup	Joel
7.3 - Arduino Setup	Michael
7.4 - Azure Setup	Michael
7.5 - Power BI	Luis
8.0 - Appendix	Remi
Document Design and Formatting	Michael, Luis
Proof Reading	Remi