(s)Nikodem Drabik, Tom Cogzell, Joeseph Troughton

Farfields  John McDermott

[Project 2003]

Contents

[Project Requirements. 2](#_Toc185244278)

[Basic Requirements: 2](#_Toc185244279)

[Our Understanding: 2](#_Toc185244280)

[Mission Statement 3](#_Toc185244281)

[Value We provide 3](#_Toc185244282)

[Application Purpose POV 3](#_Toc185244283)

[Additional Features that would benefit the client: 3](#_Toc185244284)

[User Analysis/ Possible Users: 3](#_Toc185244285)

[Contractor/Device Installer: 3](#_Toc185244286)

[Supervisor / IT administrator 4](#_Toc185244287)

[Hobbyist 4](#_Toc185244288)

[Project Roles 6](#_Toc185244289)

[Project TimeLine 7](#_Toc185244290)

[Front End Design 8](#_Toc185244291)

[Iteration One: 8](#_Toc185244292)

[Iteration Two: 11](#_Toc185244293)

[LoRaWAN and LoRA RF 13](#_Toc185244294)

[Technical Analysis 14](#_Toc185244295)

[Initial Security analysis of applications and threats: 14](#_Toc185244296)

[Important Laws: 15](#_Toc185244297)

[How it applies: 15](#_Toc185244298)

[Firebase: 16](#_Toc185244299)

[Database Low Fidelity Plan. 16](#_Toc185244300)

[Features to be used: 16](#_Toc185244301)

[To look out for: 16](#_Toc185244302)

[Proof of Concept 17](#_Toc185244303)

[Firebase Concept/Testing 17](#_Toc185244304)

[Rules for test database: Testing: 17](#_Toc185244305)

[Additional Research 19](#_Toc185244306)

[Topic - AI (Late Stage idea) 19](#_Toc185244307)

# Project Requirements.

## Basic Requirements:

A database app to manage the lifecycle of connected IoT assets from cradle to grave. As a device moves from manufacture into installation, operation, and eventual removal and recycle there is a need to record information about the device, e.g. serial number, owner, location, installation, connection, maintenance etc. The project objective is an app that records device associated information into a database to record its life history as a 'source of truth' for use by device manufacturers and business users. Emphasis on this project would be a UI experience that is more oriented towards the workflow of the lifecycle.

1)      Use of Firebase to manage data.

2)      API calls preferred to abstract out services so recommended to use microservice architecture.

3)      The project should be extendable to multiple clients running off the same database.

4)      There should be option for data to be stored on client side or client-side API keys, but functionality on cloud.

5)      Stretch goals - think scalability from 5 devices to 1000 or more.

## Our Understanding:

A structured way to keep track of devices in an IoT-based application allows for the user to monitor known statistics and update data for each device in whatever format they like. It should be flexible by allowing dynamic grouping and user-saved documents and files while also being structured enough to serve a large customer base.  – **technical aspect of Database**

A mobile interface that allows businesses and contractors to monitor and maintain their IOT devices in the field and update each device in real time. Each device should have a GPS coordinate, last serviced, and notes about the device and how it should be maintained in the future. - **implement a Real time Database**

The User Interface should be easy to understand for users of different tech literacy levels as this could be used by a manner of people tracking servers to cow trackers. **- the application of User Interface**

## Mission Statement

To provide a software solution to manage a wide range of physical devices in one simple location.

## Value We provide

With the shutdown of the Amazons IOT solution we come to fill in the market gap to provide a solution for the specific use case scenarios that our possible clients.

## Application Purpose POV

Create software that allows a business, and individuals track their IOT devices lifecycle from manufacture to installation, operation, maintenance, and eventually, recycling if possible.

Whether managing a handful of devices to scaling to thousands, our solution should be able to adapt to meet the diverse needs of clients, offering flexibility in our client-side integration as well as on our database.

### Additional Features that would benefit the client:

Maintenance scheduling

QR code scanning/reading to easily access device information.

LoRaWAN and LoRA RF communication with devices \*more information found under LoRaWAN and LoRA RF

## User Analysis/ Possible Users:

### Contractor/Device Installer:

#### Who are they

This person would be working for the company who owns the device.

The main uses of the app for this user would be:  
 Modify Maintenance Logs

Adding Devices to company database.

Removing Devices from database

#### Technical Ability

The user may not be the most technically inclined to navigate a complicated app. They can range for 18 years olds up to 65 years as it’s likely to be a manual labour position. The app must be easily teachable to all user.

### Supervisor / IT administrator

#### Who are they

This person would be managing all the devices from a centralised location like and office.

They will be in charge of ensuring all devices are up to date and if not ensure a contractor is sent to fix the issue at hand. They must have access to all devices that are in their network and no other network.

#### The main use of the app for this user?

This user will mostly be using the manage device section of the app as they are not likely to be creating new user in the device. They may be centralised in one location or be moving from one location to another so they must have remote access like a contractor to their service.

They may need to create updated or delete devices in case of issues in the real world and possible remoteness of the devices not allowing for real time updates to the database or fix human errors created by others.

#### Technical Ability

This user can be a wide range of of age however often are older due to the seniority of the position meaning that the app mist be tailored to them and their specific uses.

#### Problems this app may solve for them

An active view of most of there deployed devices. It will allow them to manage them better and be aware of their status at moment’s notice. It should fix issues of large amount of paper-based documentation that need to be looked through to find the status of a device.

#### Important features they would benefit from?

Notifications – these would provide reminders for maintenance schedules.

View of all current devices and their current stages in their lifecycle and last maintenance date.

### Hobbyist

#### Who are they

This person enjoys experimenting and utilising and messing with many electronic devices and have many devices that they need to maintain in their home for example routers, servers, NAS devices.

#### The main user of the software

They would like to keep record of all their devices they have deployed. The reason for this may be that they want to keep everything up to date with the lates driver and software updates. For this they may benefit for man app that they input data and informs them every 6 months to update their systems software.

#### Technical Ability

This type of user will be very skilled in this area as this is their passion they are likely to be much more technically inclined than the average person with tech and likely want to modify the app to be more applicable to their specific user case scenario.

#### The main issue this software would solve for them?

This software would solve the issue of maintenance scheduling of devices and life cycle of devices as all devices have a expected life cycle. This would help this users be on top of their device servicing like a MOT reminder for road users. It will help them ensure all devices are in their best shape.

# Project Roles

Tom-

**Front end/ UI/Flutter**

Task: To design and manufacture a UI for the client

Niko-

**Project Oversight, Front-end/Back-end assistance**

Task: To work with all team members to help with the deliverables. Assist in the deployment of UI and its communication with Firebase API

Joe-

**Firebase/Backend, Security of Project**

Task: To design and manufacture Firebase infostructure. Ensure that it is secure to be used by the clients.

# Project TimeLine

Project Start- October 24:

Project End- May 6:

# Front End Design

## Iteration One:

The design needs to be modular and simple. The application should be easy to use for a wide range of people who are technical. The app is going to feature 5 distinct colours for a theme.

A screenshot of a computer screen

Description automatically generated

The app's main background will be #EDF6F9 with the font colour being #303633.

For our frontend design we will be using Flutter allowing us to port to many types of devices including phone, desktop and web giving clients many ways to access the software our primary focus will be mobile support allowing contractors and admins to add new devices on site and in real time.

A screenshot of a device

Description automatically generated

Iteration two features only three colours: white, royal blue and charcoal black. I Chose royal blue as many people associate blue with technology and with the app being a tech tracker I thought it was a perfect choice of colour. The charcoal black is a less harsh black and clashes less with the colour theme.

A blue and grey rectangular shapes

Description automatically generated with medium confidence

Screens screenshot of a phone

Description automatically generated

A blue and white background with a logo

Description automatically generatedA screenshot of a phone

Description automatically generated

## Iteration Two:

After meeting with our client, we came to the conclusion for what the UI colour scheme should be. The client has a startup company “resegva” that he has designed the website for and follows the following colour scheme convention.



A screenshot of a phone

Description automatically generated

# LoRaWAN and LoRA RF

LoRaWAN

This consists of many end devices connected to gateways that receive messages using ALOHA-based protocols so that end devices do not need to be pared to specific gateways yet be within a gateway's range to send signals. These messages are then received by the network server; if it receives multiple copies of the same message, it only keeps one.

End devices can be a sensor, an authenticator, or even both. They are likely to be battery operated and connect via LoRa radio frequency modulation.

Gateways can be categorised into indoor (picocell) and outdoor (mactocell) gateways.

Indoor gateways are cost-effective and best for a location with many obstructions to the wireless signals. They have internal antennas or external pigtail antennas. Depending on the building, its possible for them to receive signals from over several kilometres away.

Outdoor gateways provide larger coverage in both urban and rural areas. Usually attached to cell towers or tall buildings.

Lastly, you need 3 servers: a network server, an application server, and a join server.

Pros -

Low power: battery-powered

Long range: in In Kilometres

Cost-effective: uses unlicensed radio frequency and low set-up cost

Scalable—can support thousands of devices on 1 gateway.

Secure \*

Cons-

Difficult to maintain custom protocols

LoRa RF is not designed for scalability

*\*Security*

LoRaWAN security

Built-in security mechanisms to protect data confidentiality, integrity, and authenticity.

Uses the AES-128 encryption for end-to-end as well as 2-layer encryption using network layer encryption and application layer encryption.

This is because its less complex, more efficient, and more cost-effective to run, which is a large point for most implementations. Yet 128 is still secure enough for current standards.

# Technical Analysis

## Initial Security analysis of applications and threats:

|  |  |  |
| --- | --- | --- |
| Risk | Affect | Countermeasure |
| User spoofing | Gain unauthorised access | 2Fa/MFA when logging in |
| Poor data base encryption | Malicious access to large quantity of data from clients stored in the database | Encrypt all data in the database to minimise the possibility |
| API attacks | Possible to infiltrate the database using illegitimate requests using the API | Ensure all requests are authorised and possibly create a rate limit on request to reduce DOS attacks on the DB |
| Malware/Ransomware | data base is taken from our control. | Frequent backups to a secondary database to minimise the threats |
| Man in the middle attack | The attacker intercepts a valid communication path between user and software and can gain access to sensitive information or inject malicious software or code into the database to gain access | Create a secure path using a multiteam of validation certificates such as TLS and SSL |
| Firebase formatting | May lead to unauthorised CRUD functionalities being performed | Format for minimum access to database. |
| File Uploads | Infected files being uploaded to the system | Ensure only authorised formats like JPG, PNG or dxcs are used and not.exe or.py to minimise the possibility of a threat |
| API key expose | Can lead to unauthorised request being sent to the firebase | Use a system that only lets authorised usage of the API port by authorised individuals. |
| Data base breaches | Data base breached by a method mentioned | To increase security, encrypt and hash all data that seems sensitive. |

### Important Laws:

There are many laws that we need to follow to ensure the project abides by the UK and EU laws, as this is where its likely to be dispatched too.

 GDPA 2018 (UK specific)

GDPR (EU wide)

Computer misuse act—to be followed by all when interacting with the software

Freedom of Information act: allows user to be aware of what data is stored on our servers about them

#### Research:

Digital economy act 2019

Network and Infrastructure Security ACT 2018

Cyber Security ACT 2019

Network and Infrastructure Security Act 2018:

Perform regular risk assessments of the application to ensure that valid measures are taking place to minimise the security risk to the network.

The ACT requires us as a company to report any cyber security breaches that may affect our systems, such as cyber-attacks, breaches into our databases or networks, any ransomware or DDOS/DOS attacks, and compromise of critical infrastructure, insider attacks, or data leaks.

A couple of possible threats will likely be around our DB usage and API ports.

### How it applies:

Overall, as a team, we will need to apply and follow all rules and regulations that apply to our database, ensuring we store data securely, perform secure connections to the database, and limit access to the lowest possible viable level as to minimise possible security vulnerabilities. A large part of the security of our infrastructure is how it will be accessed. Depending, we will need to perform some user questioning to see the ways it will be accessing the network and be able to optimise the connectivity as stated before to minimise the possibility of attacks.

# Firebase:

## Database Low Fidelity Plan.

A diagram of a software flow

Description automatically generated

## Features to be used:

* Firebase Authentication
  + Supports user sign-in and 2FA/MFA
* Cloud FireStore
  + For complex queries and and hierarchical data/scalability
* Realtime (Priority)
  + Synchronisation updating device statuses
* Cloud Functions
  + For micro service architectures like events and function execution
* Cloud Storage
  + Pictures of the installation, relevant documentation, etc
* Cloud Messaging
  + Send notification to users, like device status warnings
* Firebase Performance Monitoring/Crashlytics
  + Identify bottlenecks and problems

## 

## To look out for:

Authentication of Firebase

User/Owner login info

How to handle keys and authenticate securely in app

# Proof of Concept

Here we will be providing evidence of our proof of concept for the project.

## Firebase Concept/Testing

### Rules for test database: Testing:

Below are the rules we have implemented for our testing purposes in the database:

{

"rules": {

"users": {

"$uid": {

".read": "$uid === auth.uid", // Users can read their own data

".write": "$uid === auth.uid || auth.token.admin === true" // Users can write their own data, admins can write any data

}

},

"adminData": {

".read": "auth != null && auth.token.admin === true", // Only admins can read admin data

".write": "auth != null && auth.token.admin === true" // Only admins can write admin data

},

"publicData": {

".read": "auth != null", // Any authenticated user can read public data

".write": "auth != null && auth.token.admin === true" // Only admins can write public data

}

}

}

# Additional Research

## Topic - AI (Late Stage idea)

**AI** #late-stage

Possibility of implementing AI for assistance

**AI (adverisal learning)**

The act of manipulating an AI model to provide erroneous data or otherwise data that it should not be providing to the individual interacting with it. Machine Learning is compromised by the