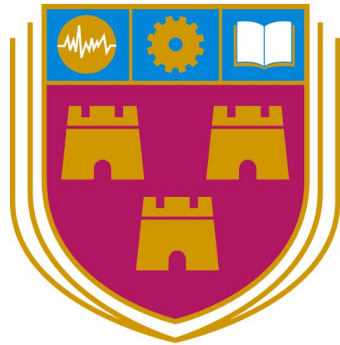


Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of*  
TECHNOLOGY  

---

CARLOW

At the Heart of South Leinster

# **LifeBuoy Monitor**

## **Final Project Report**

BSc (Hons) in Software Development

Name: Garry Byrne

Student ID: C00120055

Year: 4th Year

Supervisor: Dr Oisín Cawley

Due date: 18-04-2018

<b>Abstract</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
<b>Project Description</b>	<b>5</b>
Screens	6
Login Page	6
Register	7
Homepage	8
Homepage(marker clicked)	9
Devices Page	10
Devices Page(Search Feature)	11
Device info Page	12
Push Notification	13
<b>System Architecture</b>	<b>14</b>
<b>Problems Encountered</b>	<b>15</b>
Database	15
Course Workload/Time Management	15
Lack of Signal	16
Unfamiliarity	16
<b>What was set out to achieve</b>	<b>17</b>
Required Criteria	17
Extra Criteria	17
<b>What I achieved</b>	<b>18</b>
Low Power	18
Instant Notification	18
Plotting Device Location	19
Update Location	19
Reset Alarm	19
Movement Algorithm	20
<b>What I did not achieve</b>	<b>21</b>
Reminder	21
Update UI/UX	21
Refined Algorithm	21
<b>What I learned</b>	<b>22</b>
Firebase	22

Arduino Devices/Modules	22
Sigfox Network/IOT Device	22
<b>Do Differently</b>	<b>23</b>
Priorities	23
Hybrid App	23
Database	23
Features	23
<b>Conclusion</b>	<b>24</b>
<b>Acknowledgements</b>	<b>25</b>

## Abstract

The purpose of this document is to provide an overview of the project to any readers of this document. The document will outline learning outcomes achieved throughout the duration of the project as well as any challenges encountered through the development process.

## Introduction

This documents purpose is to look at the finished product as a whole, it will provide any readers an insight into what went into the development of the application. The document will have different sections, each detailing a different topic to do with the project, the main ones i believe are Problems Encountered, what i learned, what i would do differently and screens.

## Project Description

The purpose of this app was to create a remote monitoring solution, that would allow a user to be instantly notified if a Lifebuoy had been moved. The problem that project solved is twofold, The first problem is the most important, if a lifebuoy is not where it is meant to be in a scenario of when it is needed it can lead to a loss of life. The second is that it saves the user resources, instead of having to inspect Lifebuoy's every week or bi weekly, these inspection lengths can be increased. This might not seem to be important but some users can have thousands of Lifebuoy's that they need to inspect.

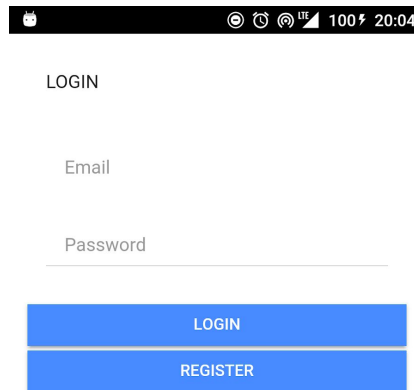
The device has to be low power to be of benefit to the user, if the device can only function for one or two weeks, there will be of little benefit to the user. With this in mind I researched low power boards and came across the Sigfox network which is specifically built for use with low power devices.

Lifebuoy Monitor was created as a Hybrid app using Ionic, this decision was made as it would suit the targeted users better ie if a user is not near their office or has no access to a computer a web app would be useless. My app allows the user to receive a notification to their smartphone once they have a connection to the internet.

## Screens

Please note when i was completing this doc i did not have the time to improve the CSS of my application, on completion of this doc my intention is to go back and improve the look of the app as it is very basic and unappealing to the eye in its current form.

### Login Page



LOGIN

Email

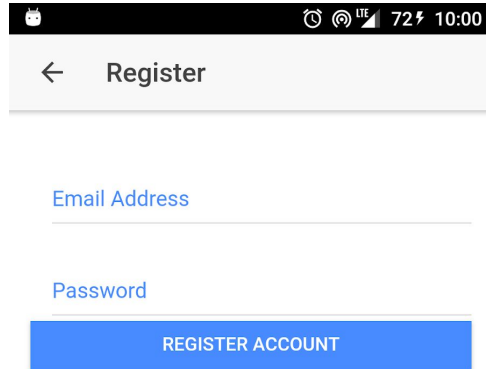
Password

LOGIN

REGISTER

The above screen shows the Login Page that appears once the user opens the app. The user is prompted to enter their email and password. Authentication is handled by Firebase Authentication. Once a user successfully authenticates they are forwarded to the next screen. If the authentication fails the user is shown a warning message telling them that they entered incorrect details.

## Register

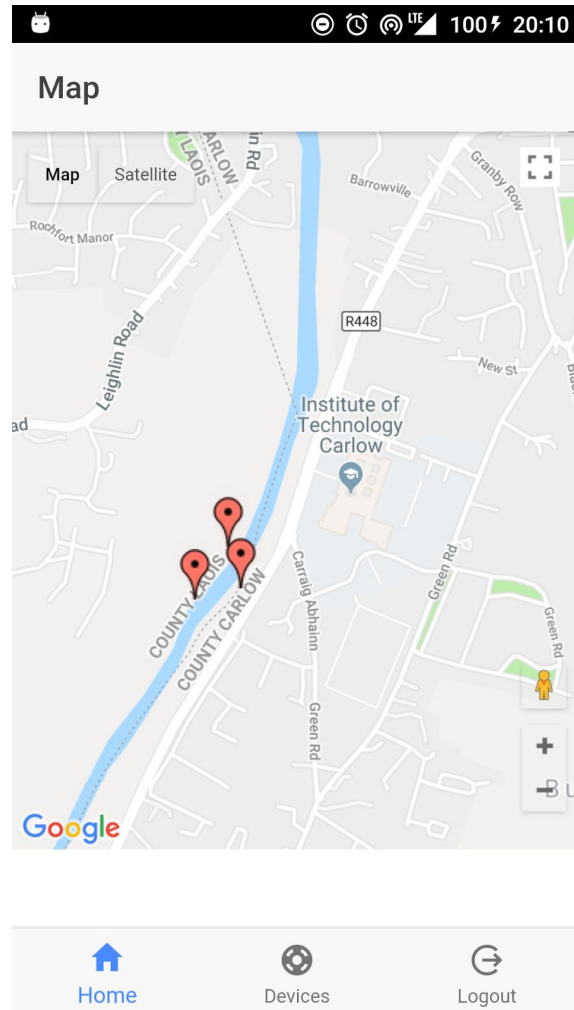


The screenshot displays a mobile application interface for registration. At the top, there is a black status bar with icons for signal, Wi-Fi, battery (72%), and time (10:00). Below this is a white header bar with a back arrow on the left and the text 'Register'. The main content area contains two input fields: 'Email Address' and 'Password', both with light blue text and underlined. At the bottom, there is a prominent blue button with the text 'REGISTER ACCOUNT' in white capital letters.

In the above picture you can see the the page a user is brought to if they need to register. The page asks the user for an email address and password. The registering process is then handled by Firebase Authentication. On successful registration the user is returned to the Login Screen.



## Homepage



The above screen shows the page a user will see once they successfully login. The map will load a google map, which will be centered on the users current position. The map will then populate a marker that represents the current location of all devices that are currently registered to the logged in user. The markers will be either green or red, green represents that no alarm has occurred for that particular device, red represents that an alarm has occurred for that particular device.

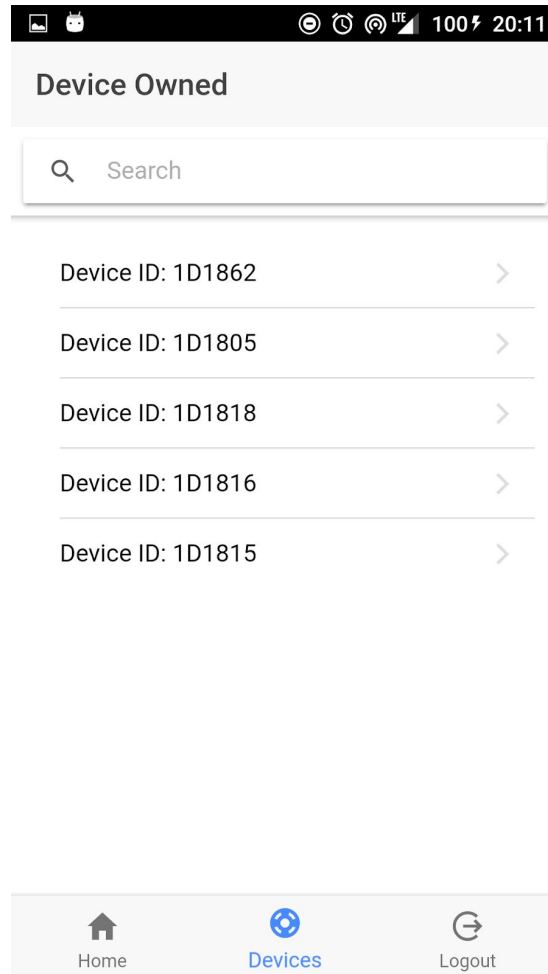
At the bottom of the page you can see a tab menu, this menu allows the user to return to the map, check on devices and to logout.

Homepage(marker clicked)



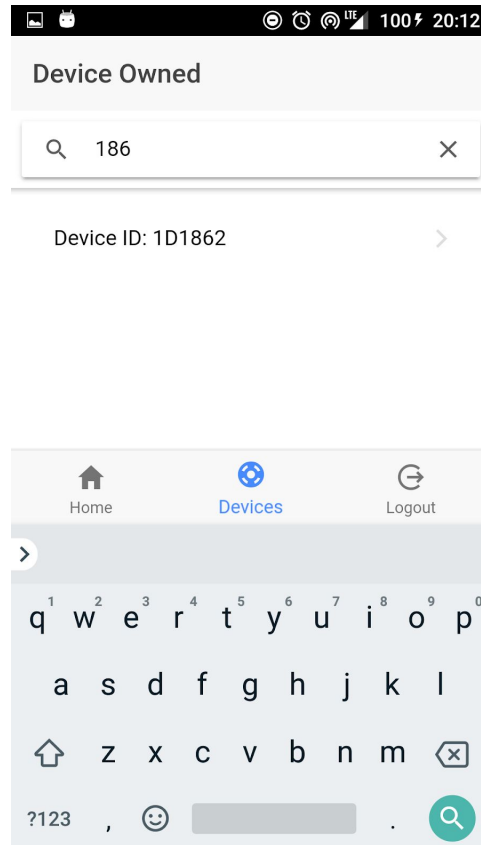
The above screen shows how the user can click on any marker and basic detail relevant to the marker will be shown eg device id,location and last inspected.

## Devices Page



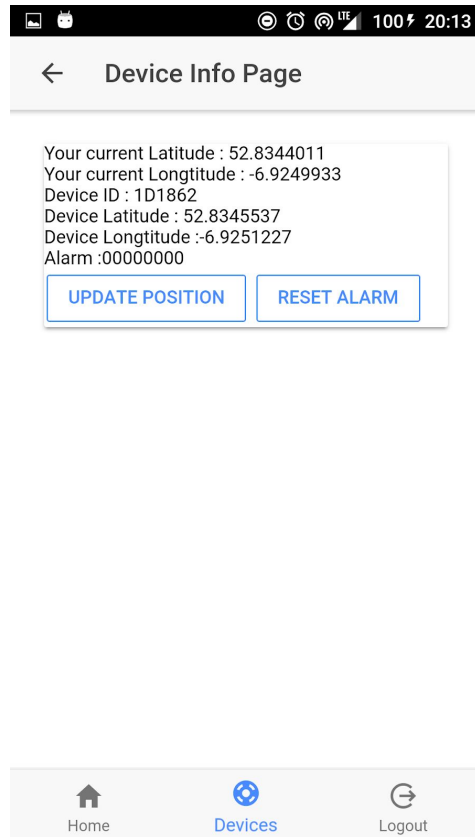
The above image shows the page the user will be brought to when they click Devices. The user will be shown a list of all their currently registered devices. The user can scroll through these devices and click on a particular device to be shown more details of that devices.

## Devices Page(Search Feature)



The above image shows the the implemented search bar, this allows the user to type in a full or partial device id. This will refine the device list in realtime. I believe this will make the searching of devices easier for the user, especially if they have hundreds of registered devices.

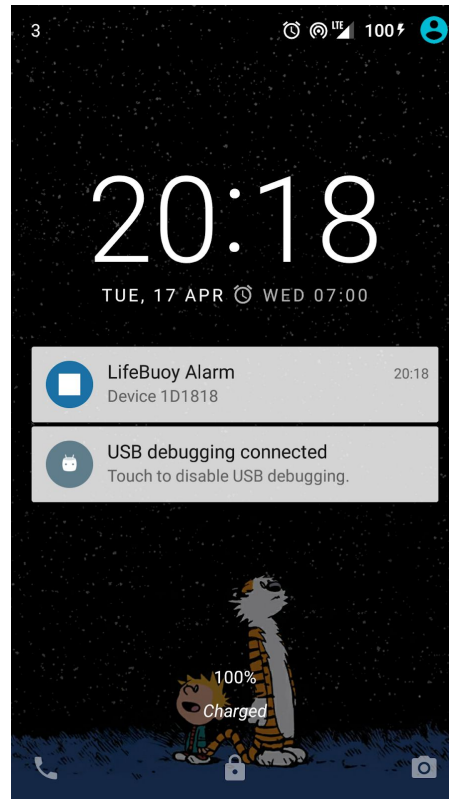
## Device info Page



The above image shows the page a user is brought to once they have selected a device from the search page. The user is shown all data stored in the database about the selected device. More information could be added to this screen if the needed eg. last inspection date.

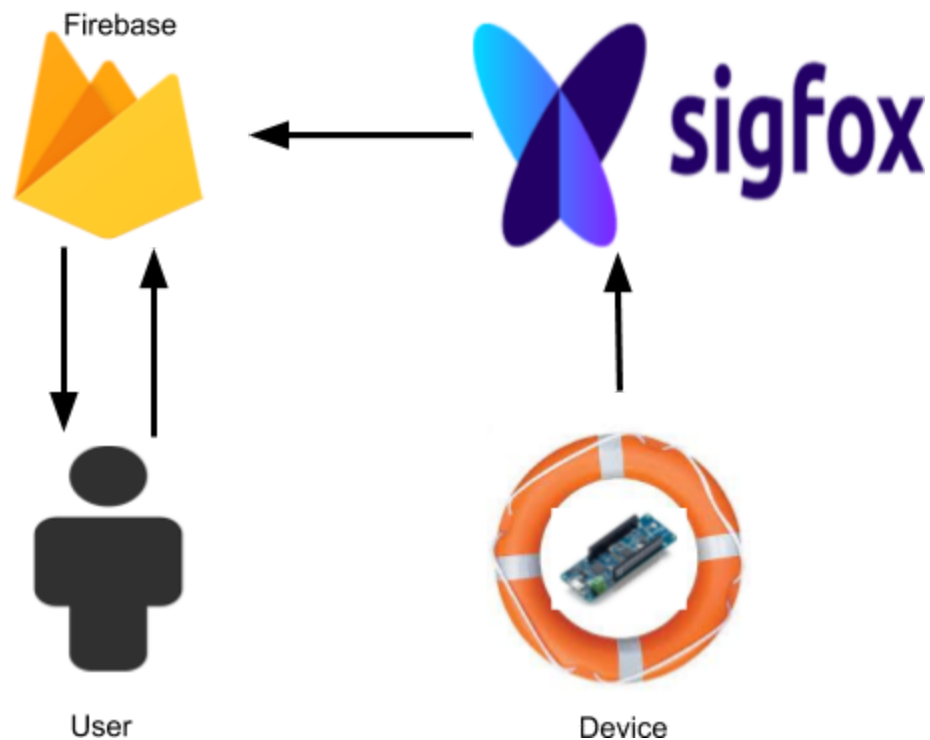
On this page the user can also update the position of the device, the position is got by using the phones internally GPS. They can also reset a device alarm from this page if desired.

## Push Notification



The above screen shows the push notification a user will receive in the event of an alarm occurring. If a user clicks this push notification the app will reopen for the user.

## System Architecture



The above diagram shows the system architecture of the application I used an arduino MKRFOX1200 as my development board which communicates through the Sigfox network. Once the data has reached the Sigfox backend the data is pushed into a Firebase Realtime Database. The user can then access the data stored inside the database through the use of the application, they can view and update specific data relevant to the user. Inside Firebase i make use of their easy to use cloud functions, these functions sort for duplicates on creation of a device or if an alarm is activated. The other function allows me to send a push notification to the user.

## Problems Encountered

During the course of this project i encountered many problems, definitely too many to list here so i will list the problems i believe had the biggest impact on me and how the project turned out.

### Database

At the start of this project i had settled on using AWS(Amazon Web Services) as my backend. I chose AWS because it had inbuilt functionality with Sigfox. Getting my data from my device to be stored in AWS turned out to be easy. I then created a website and used AWS S3 to host the website, the problem i ran into was trying to create a lambda function that allowed my website to access the data in the database. I believe the problem is on my end as i found it very difficult to follow to follow the docs. Overall i believe i wasted 3 weeks trying to get a lambda functioning correctly.

I then moved to Firebase Realtime Database which i had previous experience with. What i could not achieve in AWS in 3 weeks i got done in Firebase in less than three days. The transition from using AWS to Firebase also made me rethink my choice of creating a web app, this turned out to be a good thing as i think using a hybrid app suits my project much better.

### Course Workload/Time Management

This was a major problem for me as the course load in 4th year was much greater than expected, it seemed that every week you would have two assignments due. I am a great procrastinator, i would rather do the subjects I enjoy more than the ones i don't, while i enjoyed working on this project it always seemed there was another assignment due. It wasn't until I got my easter break that i made significant progress with my project. I believe that if i had concentrated less time on the lower credit subjects and more time on project i would have achieved more functionality in my application.

Another problem i faced was finding a dedicated spot to work from, if you set up inside the lab in the morning, you would not be able to stay there for the day. The only solution to this was to stay in the library itself but the noise level made working difficult.

If i could go back i would have just worked from home.



## Lack of Signal

This problem was twofold, originally I was using a GPS module along with my altimeter/accelerometer. From the start I had problems with configuring the module as I bought a cheap no name module off ebay. After much heartache with installing libraries I eventually got it working once, after this initial connection it never worked reliably again. I believe this would of been one of the projects stand out features if it had worked correctly.

The other problem was that i could not get a strong signal to connect to the Sigfox network from home. This hindered my progress because i could not work on the code for the arduino at home. Many weekends i would get into my car and drive 2 minutes down the road just to test a small amount of code. After christmas i had enough of this so i went through the entire house and found I one location that allowed me to connect reliably to the Sigfox network.

## Unfamiliarity

Another major problem i faced was the unfamiliarity i had with the arduino board and connecting the modules. I had never used an arduino or soldered pins to a board, this led me to spending two weeks learning how to solder and connect modules correctly. I also learned about the fragility of these devices as i snapped 2 antenna wires.

## What was set out to achieve

As I didn't create the spec of this project myself i was tied to meeting certain criteria. I will list the criteria I wanted to achieve below, with a small explanation of each one.

### Required Criteria

- Low Power - The project had to have low power usage so as to allow the device to function for an extended period of time in the wild.
- Instant Notification - If the device senses movement the user is instantly notified.
- Plotting Device Location - Allow the user to view their devices on a live map.
- Update Location - Allow the user to update the position of the device.
- Reset Alarm - If an alarm was to happen, the user should be able to reset the alarm for the specific device.
- Movement Algorithm - Algorithm for the sensing of device movement.

### Extra Criteria

- Reminder - Inspection reminder for the user
- Improved UI/UX - Improved on the basic look of the device.
- Refined Algorithm - Refine the algorithm for greater accuracy

## What I achieved

### Low Power

I achieved power savings within my device by using libraries that allowed me to put the mkrfox1200 into sleep mode. In this mode the device listens to a specific port for any activity, once this activity occurs the device will be woken. The activity that wakes the device in my situation is movement which is sensed using the altimeter/accelerometer. This code can be seen in my Technical manual along with comments describing the code.

### Instant Notification

I achieved Instant Notification within my device by using the cloud functions provided by Firebase. The created function listens for any changes within the the devices table, specifically on the alarm attribute. Once this change has occurred the user will be notified by push notification.

The push notification works by getting the FCM token that is tied to a specific device and user on initial starting of the app. This FCM is stored in a separate table within firebase along with the users email address which is used to identify what devices belong to which user. Usually a device receives a push notification within 10 seconds of the alarm being updated.

This code can be seen in my Technical manual along with comments describing the code.

## Plotting Device Location

I achieved the functionality that allows the user to visually see the location of their devices. I implemented this by using Google's, google maps api. Once the user logs in they are shown a map with all their devices displayed on it, the map is centered on their current location for convenience. The markers are displayed by pulling the relevant devices from the database and looped through to get the individual latitudes and longitudes. They are then displayed as either green(no alarm) or red(alarm) pins. This code can be seen in my Technical manual along with comments describing the code.

## Update Location

I achieved the functionality that allows the user to update the location of the device based on their current location. This has two benefits, the first is that when the device is first activated it does not have the precise GPS location, this is because the Sigfox network gets the distance to the nearest transmission antenna and uses the signal strength as a way to calculate the location, this is a very inaccurate way to calculate the location of the device. So allowing the user to update the location by using their phones inbuilt GPS provides the device the exact location. On update of the device the map will reload with the updated device displayed in the correct location.

This code can be seen in my Technical manual along with comments describing the code.

## Reset Alarm

I achieved the functionality that allows the user to reset the alarm remotely. The benefit of this is that it allows the user to just reset the alarm without having to restart the device. Once the devices alarm is reset the marker will reload on the map and be displayed into the correct color.

This code can be seen in my Technical manual along with comments describing the code.

## Movement Algorithm

I achieved the functionality that allows the accurate sensing of movement. The benefit of this is that it allows some movement of the device without setting off the alarm. This was accomplished by the taking of sensor data repeatedly, this was used to improve the algorithm. I done this by placing the device 1.5m above the ground by hook, replaced repeatedly. This allowed me to make slight adjustments to the algorithm..

This code can be seen in my Technical manual along with comments describing the code.

## What I did not achieve

### Reminder

An extra criteria i wanted to implement but did not get around to implementing was a function that allowed the user to set the date they last checked a Lifebuoy. The app would then send a push notification to the user informing them about the upcoming due inspection. I did not get to implement this feature as i ran out of time.

### Update UI/UX

An extra criteria i wanted to implement but did not get around to implementing, was to update the applications CSS as right now it is very basic, as well as this I would have liked to have had time to use the app in depth so that i could refine the user experience to a more acceptable level.

### Refined Algorithm

An extra criteria i wanted to implement but did not get around to implementing, was to continually refine the movement algorithm. The current state of the movement algorithm works well, but i feel with a bit more time i could improve on it hugely.

## What I learned

### Firestore

Although i had past experience with using Firestore, I still learned a lot about Firestore-Cloud functions and FCM(Firebase Cloud Messaging). Without the use of cloud functions I believe I would not of been able to complete the functionality in what I wanted to achieve while completing this project.

The only issue I had with Firestore cloud functions was that you had to write the code in a selected IDE and then upload it to Firestore using the firestore CLI, I found this very annoying when I was trying to test the code as it was very time consuming.

### Arduino Devices/Modules

This it the topic that i learned the most in, from connecting the device to its sensors, to soldering wires to their corresponding pins. It also gave me experience in using the Arduino IDE and using their included libraries. I had experience in coding C++ from second year, this helped me greatly in coding for my device.

### Sigfox Network/IOT Device

Before this project i had never really heard of IOT devices and networks like Sigfox/Lora. The Sigfox network is designed to allow long range communications at a low bit rate and low power consumption which means that a device can remain active for months with just a battery for power. To me this was mind blowing as it had the belief that if you had to transmit data you would need a more reliable power source than batteries. But from using this technology I now realise that these devices need very little power to function correctly.

Also from researching and actually using this technology i can see that this is really the future in terms of programming. Everything can be connected to a IOT device so there is countless opportunities in the future for jobs.

## Do Differently

There is not many things that i would do differently if i could restart the project.

### Priorities

If starting again i would of allocated more time to actually programming the project instead of spending so much time researching technologies that i already knew wouldn't suit the project. Also i would of concentrated less time on lower weighted subjects and spent that time working on the project.

### Hybrid App

If starting again i would not have spent time developing a webapp, I would of went straight for a Hybrid app. A hybrid app to me seems like a solution that can fit most specifications of apps. Once you get over the learning curve of using Ionic and Cordova the process of creating a app is very straight forward. Hybrid apps can have most of the functionality that native apps.

### Database

If starting again i would of stuck to the database that i knew and had experience with instead of wasting time trying to learn how to use AWS(Amazon Web Services) correctly. I still believe that AWS would of suited me better if i could have got it working correctly. If the AWS docs were more easy to understand and follow i think i would of stuck to AWS.

### Features

If starting again i would of set myself a timeline to get features done by. I spent too long trying to get certain features working even though it was to the detriment of other features. In the end i got the functionality i wanted to achieve completed but i think setting timelines would of caused a lot less hardship.



## Conclusion

Overall I feel that I have accomplished this objective to a satisfactory level, i believe that i have reached the majority of my wanted criteria, maybe the code isn't elegant and the CSS is not great but i believe that i have produced a functional app. Over the course of the project i have had to give two presentations(something that i hate doing), went through 4 iteration cycles(barely). It is a experience that i hope i won't have to repeat in such a short amount of time again.

## Acknowledgements

I would first like to thank my close circle of friends who not only put up with my ramblings but always offered a different perspective of any of my problems.

In addition, I would like to thank my supervisor, Dr. Oisin Cawley, for his valuable guidance. Oisin whom, created the initial specification of the LifeBuoy Monitor but still encouraged me to develop the application in my own personal style.