

Boxing Analyser

Project Engineering

Year 4

Oisin Concannon

Bachelor of Engineering (Honours) in Software and Electronic Engineering

Galway-Mayo Institute of Technology

2020/2021



**Declaration**

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) in Software and Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgements**

I would like to thank all the lecturers for helping me achieve this project and guide me on the correct path to completing this project successfully. A special thankyou to Paul Lennon and Brian O Shea as these two played a big part in the guidance of this project. Without them this would not have been achieved and I would like to send my sincerest thankyou to everyone who played a part.

**Table of Contents**

[1 Summary 6](#_Toc71929437)

[2 Poster 7](#_Toc71929438)

[3 Introduction 8](#_Toc71929439)

[4 Project Background 9](#_Toc71929440)

[4.1 Martial Arts 9](#_Toc71929441)

[4.2 Front End Development 10](#_Toc71929442)

[5 Project Architecture 11](#_Toc71929443)

[6 Project Plan 12](#_Toc71929444)

[7 React Native 13](#_Toc71929445)

[7.1 React Navigation 13](#_Toc71929446)

[7.2 Picker 15](#_Toc71929447)

[8 Accelerometer 16](#_Toc71929448)

[8.1 How it works? 17](#_Toc71929449)

[8.2 Accelerometer Integration 18](#_Toc71929450)

[9 Charts 20](#_Toc71929451)

[10 NodeJS 21](#_Toc71929452)

[11 Ethics 22](#_Toc71929453)

[12 Conclusion 23](#_Toc71929454)

[13 Appendix 24](#_Toc71929455)

[14 References 25](#_Toc71929456)

# Summary

As part of my final year studying BEng Software and Electronic Engineering, I was required to complete a project engineering module. For this project I came up with “Boxing Analyser”. The goal of this project was to attempt to solve and ongoing issue for amateur fighters. As a fighter myself I find it difficult to monitor statistics and analyse previous bouts. This was the inspiration behind the project as I am passionate about combat sports and have a fascination with the constant improvement aspect.

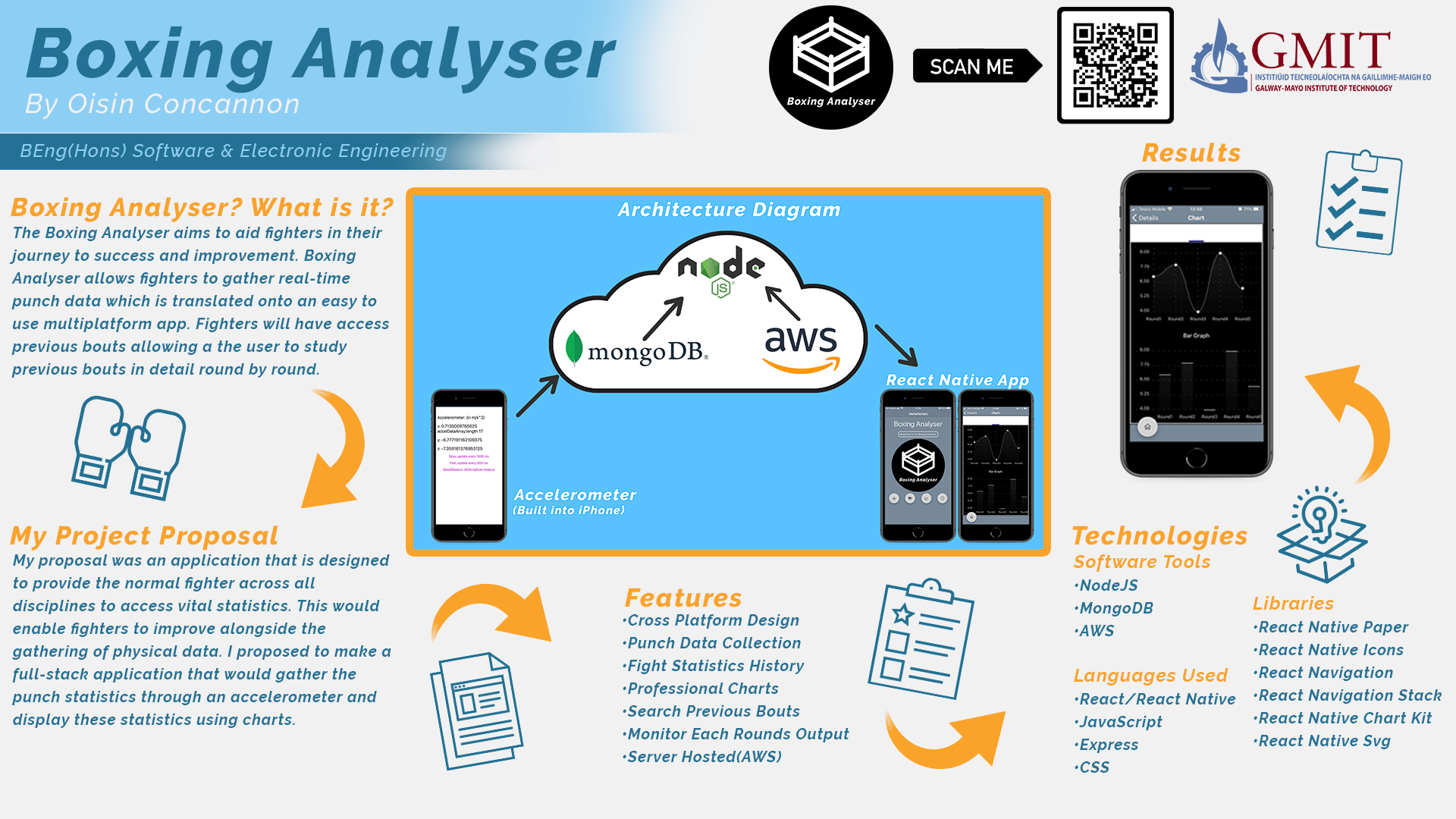
This module took place over both semesters and this project came to fruition in the very first week of the first semester. The aim of “Boxing Analyser” was to provide users with real-time statistics as well as the ability to study previous bouts. It was planned to generate charts based on the punch output each round. This would give the fighters or coaches’ real-time statistics visually in a practical way. The approach was simple, I researched similar products on the market and I found very little. After this I was forced into a lot of trial and error. I had a very clear image of what I wanted to achieve and once the vision was clear I just had to figure out the path to get to the result.

I originally planned to make a full stack web application using the framework “Express” however I was guided away from this route by my supervisor. I settled on creating a cross platform application using React Native. This would give my application versatility and would be compatible with all operating systems. React Native was a key part in the completion of this project as I was able to gather a strong knowledge on the framework quickly.

As planned I completed the project and my application is gathering real time statistics and pushing them to my database. I also have a feature which allows fighters to search previous bouts and view the charts of that specific bout.

In conclusion, I was very satisfied with how this project transpired. If I could plan my next step for this particular project I would working on the accelerometer side and start to detect specific punches and chart these punches whether it’s a straight punch, hook or uppercut.

# Poster



# Introduction

Firstly as an introduction to this report I would like to introduce myself. My name is Oisin Concannon and I am a final year student studying Software and Electronic engineering. As part of this final year we are tasked with the design of a final year project. Due to my immense interest in martial arts I decided to base my project on this topic as I feel I have the knowledge and the passion to design something useful.

This was a yearlong module and it was over the duration of seven months. Over the course of two semesters I have put together the “Boxing Analyser”, this project is very close to what I had envisioned at the beginning of the year. With some more time I would make some useful additions to make the application even more feature based and user friendly.

The aim of this project was to provide a simple but professional user interface that would allow martial artists to gather real tine data with the ability to retrieve previous bouts and study them in greater detail. Boxing Analyser solves these issues and provides physical data to fighters.

Its ease of use is vital as this application does not require the bells and whistles of much more complex projects. Using react native I created a very practical user interface. React Native was the backbone for the application. MongoDB provides the database aspect of the project allowing fighters to store data into columns and retrieve them at a later date if they wish. Lastly, the project is hosted using Amazon Web Services (AWS). Using AWS allows me to share the application with others and give them access to the client side of the application. This entire project was planned using Microsoft Project and was regularly used as a reference.

In this report I will discuss in greater detail, the scope of this project and the timeline as well as my approach and problem solving.

# Project Background

## Martial Arts

Martial arts is not as popular as some mainstream sports and people can often be misinformed on the subjects. Hopefully in this section I can give more of an insight into the inspiration and the background for this project.



Figure 4‑1 My First Bout

I have been training in the discipline of kickboxing for four years and I have become obsessed with learning and improving constantly. Due to this passion for improvement and the opportunity to complete a final year project, it granted me the chance address this topic. I was presented with the opportunity to assist fighters in terms of their personal improvement and development as a fighter.

## Front End Development

As I have developed over the duration of this course, I have slowly gravitated towards the front end aspect of software development. I feel there is a more visual and interactive characteristics. Due to this I chose to design a full stack application using MERN stack (Mongo, Express, React, Node).



[1] Figure 4‑2 MERN Stack

React Native provided a professional framework that had a smooth learning curve. It enabled me to develop a cross platform application. This is a massive advantage as my application can appeal to users across all platforms. A major advantage also in the selection of React Native is the ability to only create one project that can function across all platforms. This is highly efficient and aids developers in the creation of such applications.

# Project Architecture

Below is my architecture diagram. As can be seen, an accelerometer sends data to the cloud. This data is handled by NodeJS which is hosted on AWS. NodeJS then sends this data to my React Native application. The data across the five rounds is gathered and stored in MongoDB alongside some user details such as name, opponent name and date. These bout details can be retrieved at any point using the search feature. The punch data across five rounds is then used to spin up charts as can be seen on the right hand side.

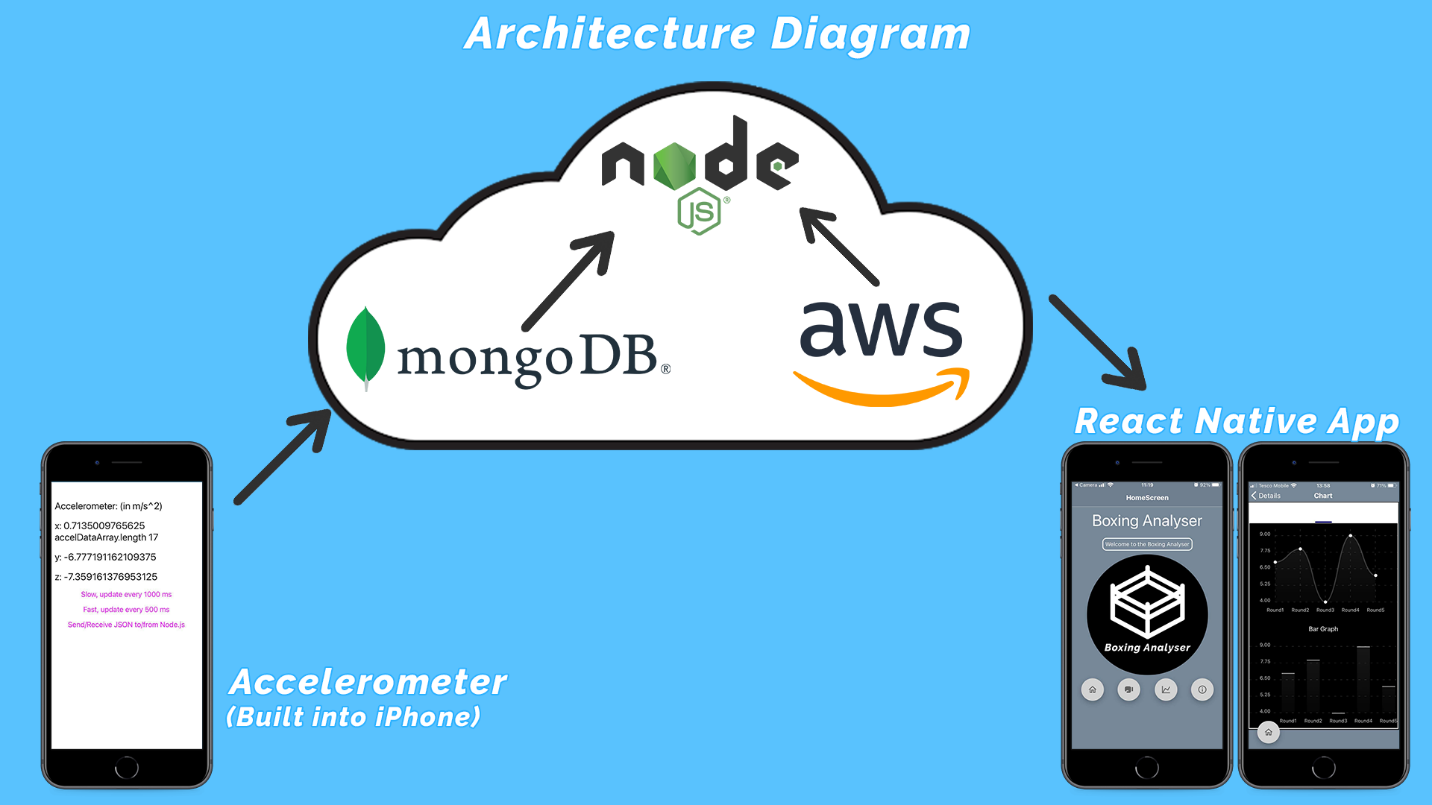


Figure 5‑1 Architecture Diagram

# Project Plan

# React Native

In this section I will discuss in detail the use of React Native and the reasoning about using this approach. I will break down some of the key features that were used with regards React Native.

“React Native combines the best parts of native development with React, a best-in-class JavaScript library for building user interfaces” [2]. This was one of the driving forces behind this viewpoint. The ability to create native applications is a massive advantage that suited the style of project I was creating.

## React Navigation

React Navigation is a very functional way of moving between screens in an application. “React Navigation provides a straightforward navigation solution, with the ability to present a common stack navigation and tabbed navigation patterns on both Android and iOS” [3]. Firstly I was required to make a stack, this is a common container for all the screens required. As can be seen below, we have created a navigation container and then inside this we declare our screen. I have only used one screen in this example called “HomeScreen”.



Figure 7‑1 Stack Navigation Container

The component name needs to be identical to the import component name to allow me to use a multi file structure. As can be seen in “Figure 7-2”, the component that is being imported is identical in name to the component in the stack navigator and this allows me to import functions/screens from separate files keeping my “App.js” (or main file) clean and clutter free. Having navigation working in a multi file format proved quite tricky and it was a problem towards the end of the timeline however I realised the issue and luckily it was the correct solution. It was difficult as there was nothing similar in any search engine results and there was very little detail out there on the web.



Figure 7‑2 Component Import

I also provide a route to the screen. This route is used when I require to move to another screen. Below I will provide an example and explain how this functions.

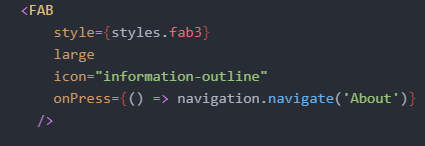


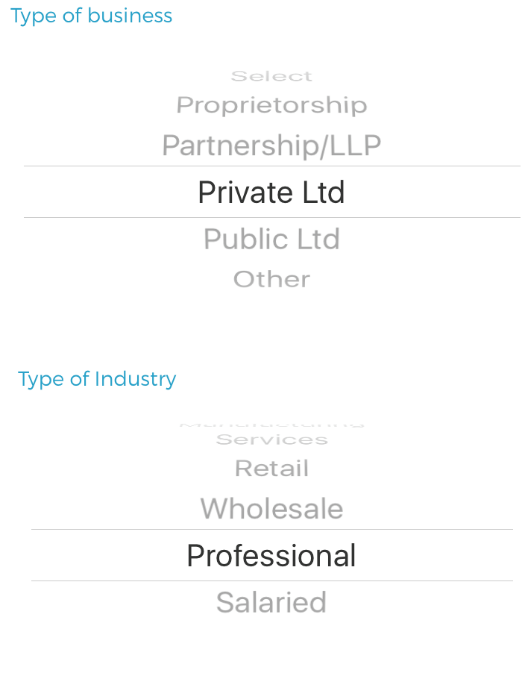
Figure 7‑3 Navigation Route

When an action is performed, in this case its “onPress” we call the function “navigation.navigate()” and we pass it the route and in this example the route is “About”. This “FAB” [4] when pressed will navigate to the route “About”. It’s a very simple yet effective method of moving throughout various screens in react native applications

## Picker

One of the first components I implemented into my project is a picker. This is the equivalent of a dropdown list however it has a bit more complexity to it.

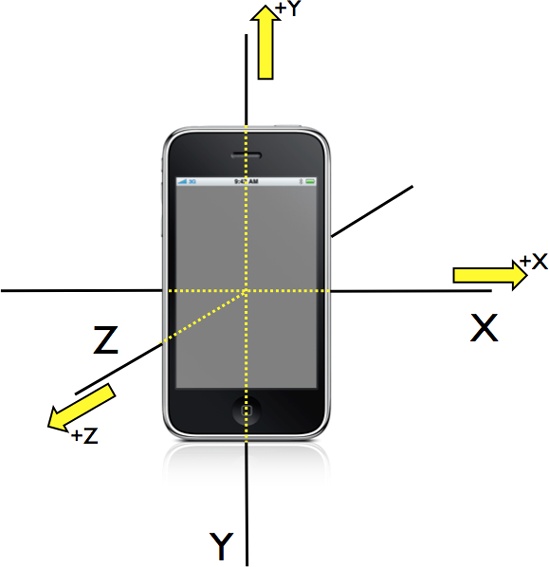
The picker looked particularly impressive on the iPhone as can be seen from the image below.



[5]Figure 7‑4 Picker Example (iPhone)

The iPhone picker is similar to a wheel and when scrolling through the items it spins them in a wheel like motion. This component is a standard react native component which I availed of in two areas of my project. I used a three element picker to select the date for each bout to give extra detail when accessing this information at a later date. I also used an additional picker when searching the database for previous bouts. When the name is search it fills the picker with the dates of the bouts retrieved from the “fetch API”.

# Accelerometer

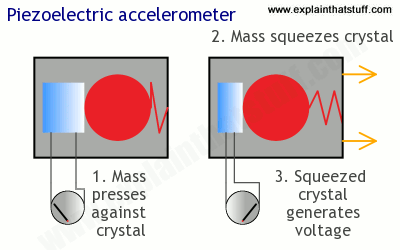
The Accelerometer is a key piece of equipment with regards to this project. As I did not receive my physical parts this presented me with a problem. I was forced to view the challenge from a different angle. With some guidance from my supervisor, we came to the conclusion that harnessing the accelerometer in my personal phone would be the quickest approach to solve the issue as time was lacking.

[6]Figure 8‑1 Accelerometer Axis

As can be seen in “Figure 8-1”, there are three axis. X, Y and Z. In my hand I turned the phone and focused on the X axis. When the reading overcomes a certain threshold I would increment the punch count. This is not the most accurate proposal however due to lack of time and resources I was limited to this outcome.

## How it works?

“An accelerometer is a device that measures the vibration, or acceleration of motion of a structure. The force caused by vibration or a change in motion (acceleration) causes the mass to "squeeze" the piezoelectric material which produces an electrical charge that is proportional to the force exerted upon it”[7]. This quote was taken from the website “omega.com” and explains the concept of an accelerometer in a concise fashion.



[8]Figure 8‑2 Accelerometer Diagram

An accelerometer measures the acceleration in its relevant axis. It calculates the vibration of the movement, therefore the higher the vibration the more sever the acceleration. I have supplied a diagram above showing how a “Piezoelectric accelerometer” functions. The mass of a particular movement presses against the crystal inside the accelerometer, this mass squeezes the crystal. The more mass supplied would indicated that the accelerometer would produce a higher voltage value. Concluding the higher the mass the more output voltage will be generated. This is very clear as the quicker you accelerate the higher the output value will be.

## Accelerometer Integration

Finally I would like to discuss how I implemented the accelerometer into my project. With the guidance of my supervisor we designed a very basic application that would harness the accelerometer in a phone. It was not ideal however it does function and executes everything we need.

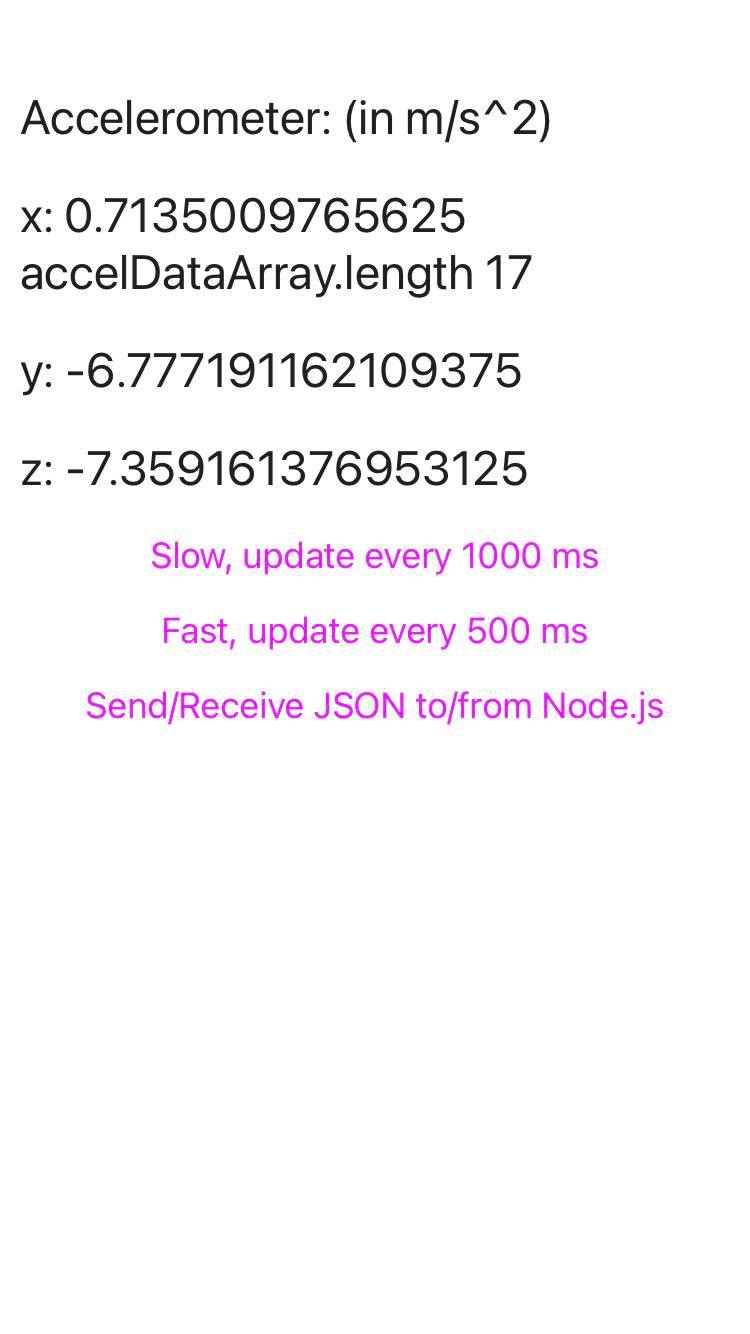


Figure 8‑3 Accelerometer Application

The data gathered from the accelerometer is sent to NodeJS via a JSON object. This is stripped down by NodeJS to extract the value of the accelerometer reading. This data is handled by NodeJS and a punch is determined on whether or not the individual movement exceeds the threshold. If the threshold is exceeded the punch count is incremented. This count will be incremented every time the threshold is exceeded and will be stored and reset to zero when the round has ended and the user has indicated that by pressing the stop round button.

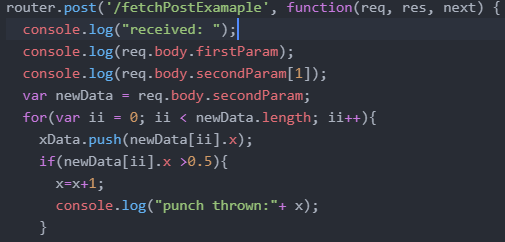


Figure 8‑4 Accelerometer Data Handling Code

As can be seen in “Figure 8-4”, we receive the data in JSON format. We extract the body and add it to the three element object(X, Y, and Z). We then extract the X value from this newly created object and check if it has surpassed the threshold of “0.5”. If this is the case we increment x which is our punch total for that round. When a request is sent back to NodeJS indicating that the round has been terminated, the value is stored and then reset for the next round.

# Charts

Charts were a vital feature in this project. Clear and concise charts were required to provide the user with visual data charts. This allows for quick analysis and even by observing curves and spikes, it can provide an effortless analysis.

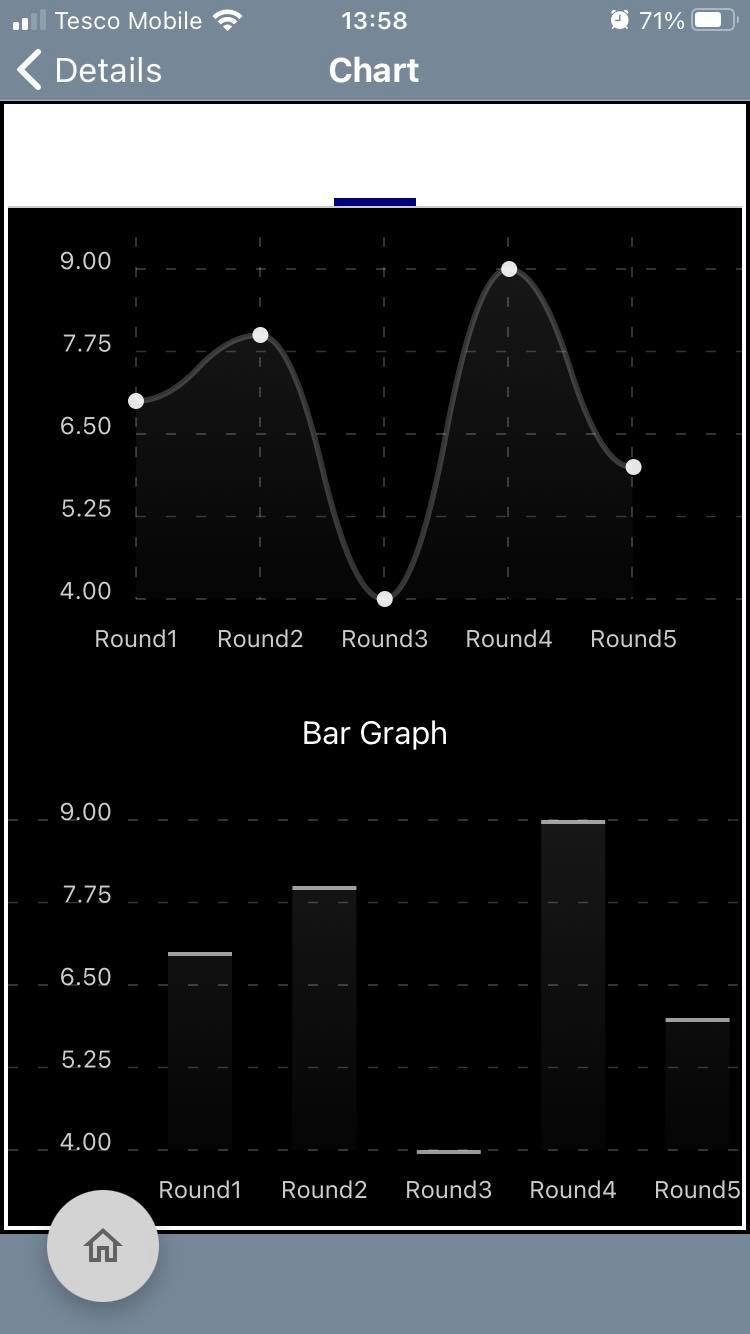
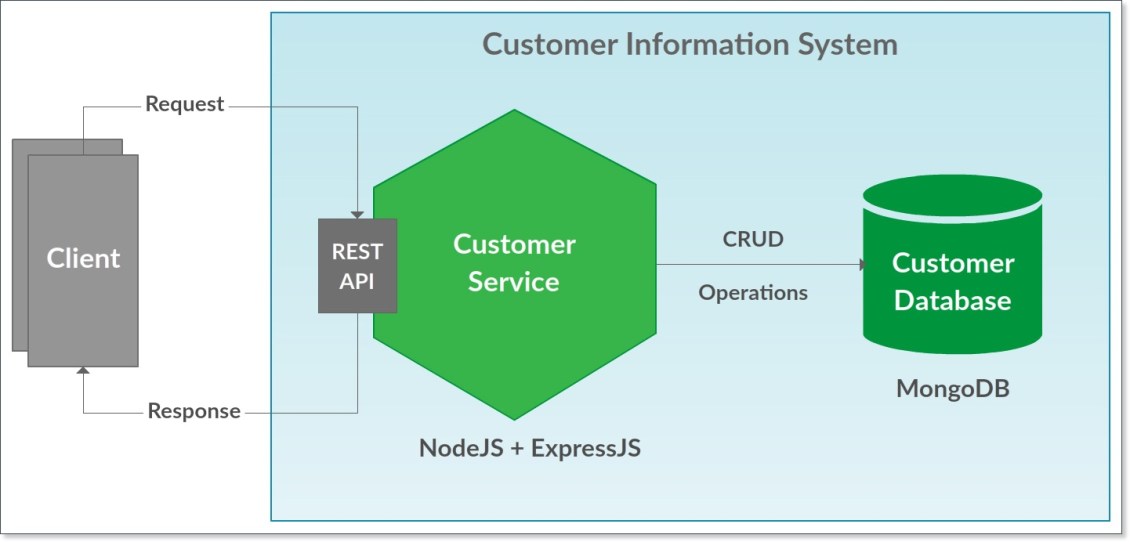


Figure 8‑5 React Native Chart Kit Example

As can be seen in the image above, this is the chart output in my application. I provide the label for the axis and then I pass it an array containing the punch totals for each round. I generate a line chart and also a bar chart. I felt that these two graphs were most adequate and suited the style of data being gathered. The react native chart kit isn’t as impressive on the web however it is extremely professional on iPhone. Some of the charts available are, Pie charts, Contribution graphs as well as the two I have previously mentioned and used in my application.

# NodeJS

“Node. js is primarily used for non-blocking, event-driven servers, due to its single-threaded nature. It's used for traditional web sites and back-end API services, but was designed with real-time, push-based architectures in mind” [9].NodeJS supplied the building blocks for this project. It is used to handle request by the user. The user will send requests to NodeJS and then NodeJS will handle these event-driven requests and act appropriately.



[10]Figure 9‑1 Rest API requests example

I have provided an example of requests being sent to NodeJS and how the rest API’s are handled. The server is event-driven

# Ethics

Include a short section on ethical considerations in your project or in the field of study of your project.

# Conclusion

Write a short conclusion. What is the outcome of the project? Perhaps you have a product prototype, or some results, or a demonstratable system.

Do not use your conclusion to tell the reader what you might have done if you had more time, but keep it focussed on what you actually have done. You can mention future opportunities for further development of the work, but keep this part short.

# Appendix

AWS – Amazon Web Services

FAB – Floating Action Button

API – Application Programming Interface

JSON – JavaScript Object Notation

# References

[1] MERN Stack Image: https://csharpcorner.azureedge.net/article/what-is-mern-stack/Images/The%20MERN%20Stack.jpg

[2] React Native: <https://reactnative.dev/>

[3]React Navigation: https://reactnative.dev/docs/navigation#:~:text=If%20you%20are%20getting%20started,on%20both%20Android%20and%20iOS.

[4] React Native FAB: <https://callstack.github.io/react-native-paper/fab.html>

[5] Picker Example Image: <https://i.stack.imgur.com/RppGo.png>

[6] Accelerometer image: https://www.oreilly.com/library/view/basic-sensors-in/9781449309480/httpatomoreillycomsourceoreillyimages873889.png.jpg

[7] Accelerometer by Omega: <https://www.omega.com/en-us/resources/accelerometers#:~:text=An%20accelerometer%20is%20a%20device,the%20force%20exerted%20upon%20it>.

[8] Accelerometer Diagram Image: <https://cdn4.explainthatstuff.com/piezoelectric-accelerometer.png>

[9] Why use NodeJS by toptal: <https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>

[10] NodeJS request example image: <https://i2.wp.com/novicedeveloper.com/wp-content/uploads/2018/11/Architecture-Customer-Information-System-Horizontal.jpg?resize=1130%2C542&ssl=1>