
# Boxing Analyser

Project Engineering

Year 4

Oisin Concannon

Bachelor of Engineering (Honours) in Software and Electronic Engineering

Galway-Mayo Institute of Technology

2020/2021


**Figure 1 Application Homescreen**

# 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 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 who helped guide me on the correct path to completing this project successfully. A special thank you to Paul Lennon and Brian OShea, as these two played a big part in the support of this project. Without them, I would not have achieved this, and I would like to send my sincerest thank you to everyone who played a part and helped in any form.

# Table of Contents

# 1 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 an ongoing issue for amateur fighters. As a fighter myself, I find it challenging 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 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 of 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 transparent 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, my supervisor guided me away from this route. 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 crucial part of completing this project as I gathered a strong knowledge of 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 that allows fighters to search previous bouts and view the charts of that specific bout.

In conclusion, I was delighted with how this project transpired. If I could plan my next step for this particular project, I would be working on the accelerometer side and detecting specific punches and charting these punches, whether it’s a straight punch, hook, or uppercut.

# 2 Poster

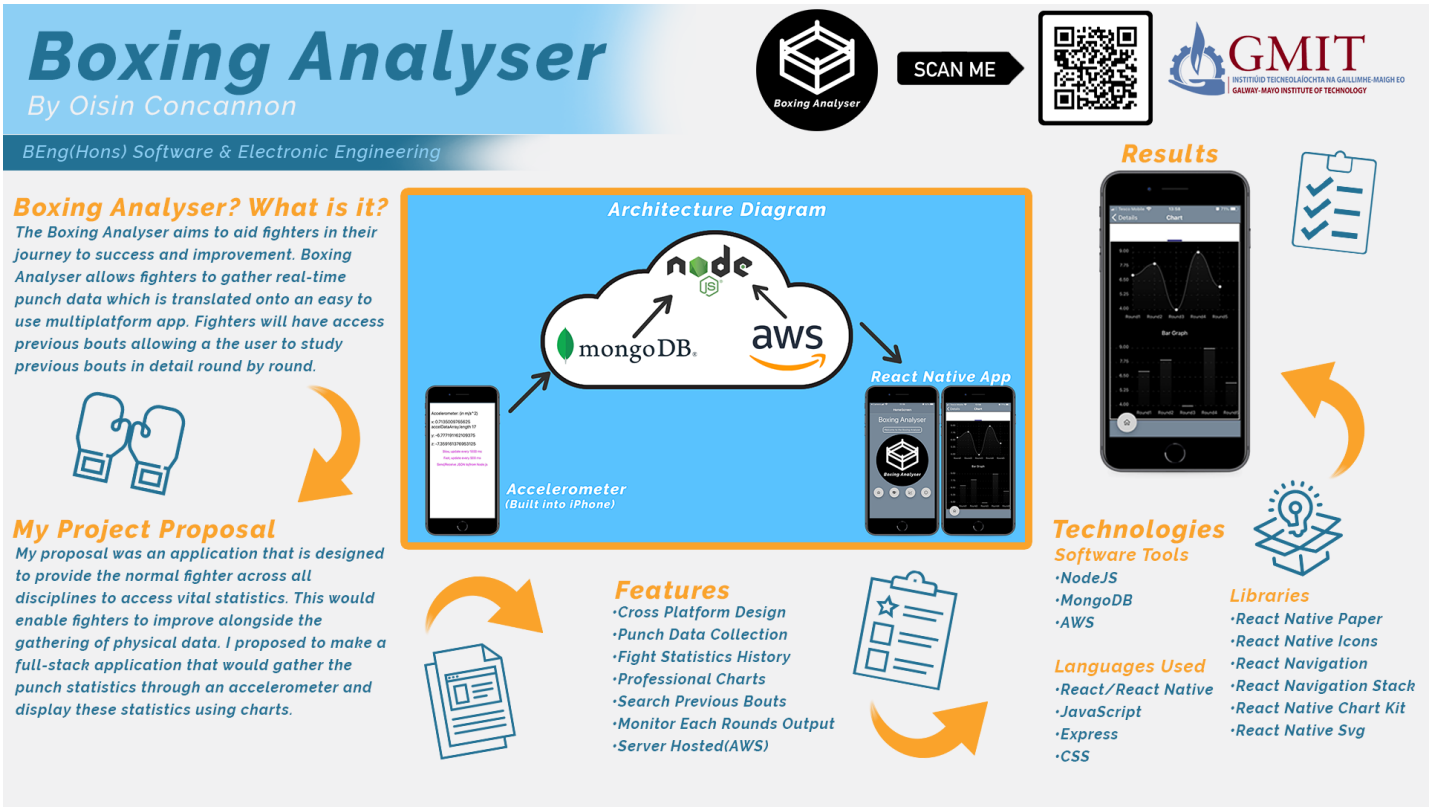



**Figure 2-1 Project Poster**

# 3 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 for designing something useful.

This was a year-long module, and it was over seven months. Over 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 valuable additions to make the application even more feature-based and user friendly.

This project aimed to provide a professional but straightforward user interface that would allow martial artists to gather real-time 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 an efficient user interface. React Native was the backbone of 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. NodeJS is being used as an event-driven server to handle the requests of the user. Lastly, the project is being 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 the scope of this project and the timeline in greater detail, as well as my approach and problem-solving.

# 4 Project Background

## 4.1 Martial Arts

Martial arts is not as popular as some mainstream sports, and people can often be misinformed on the subject. Hopefully, in this section, I can give more 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. This passion for improvement and the opportunity to complete a final year project allowed me to address this topic. I was presented with the opportunity to assist fighters in their progress and development as a fighter.

## 4.2 Front End Development

As I have developed throughout this course, I have slowly gravitated towards the front end aspect of software development. I feel there are more visual and interactive characteristics. Due to this, I chose to design a full-stack application using the 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 significant advantage in selecting React Native is the ability to create one project that can function across all platforms. This is highly efficient and aids developers in the creation of such applications.

# 5 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 shown on the right-hand side.

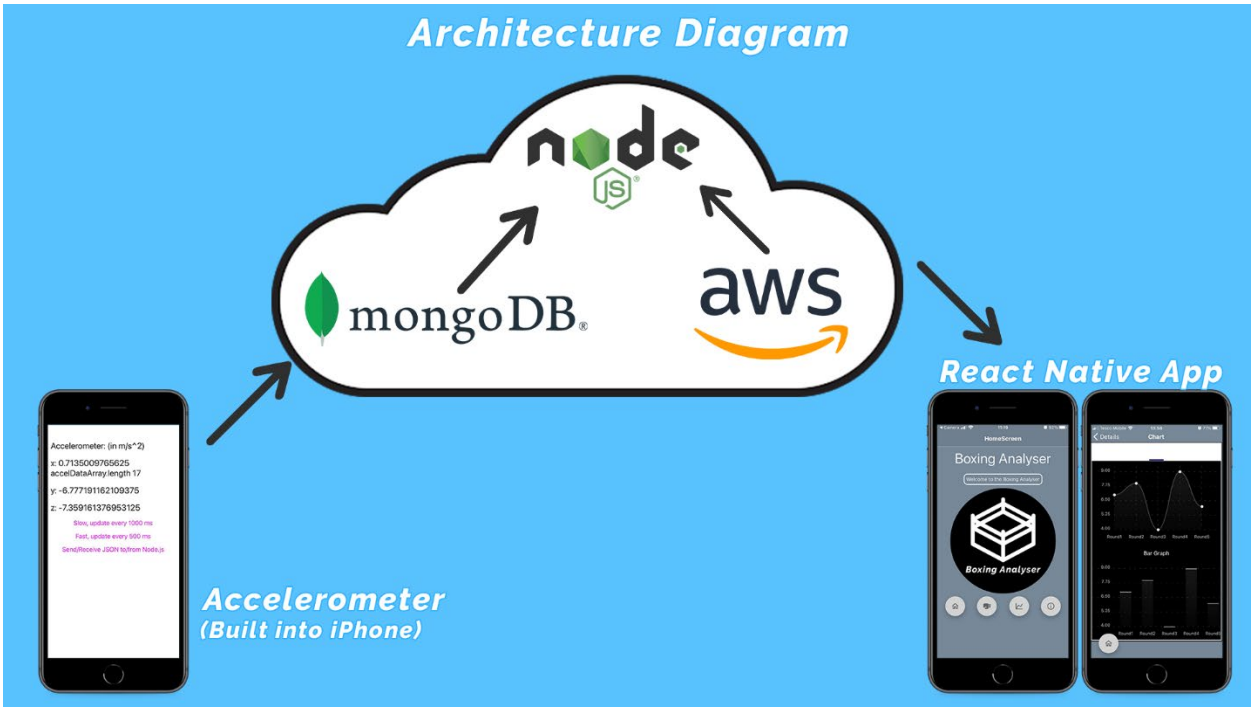
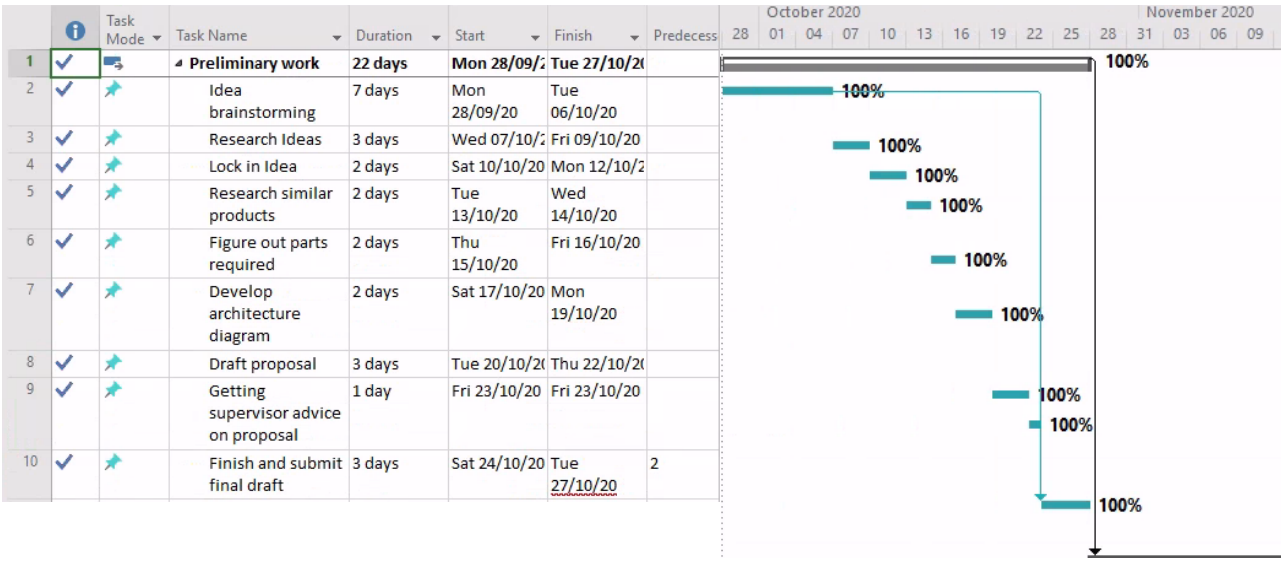



**Figure 5-1 Architecture Diagram**

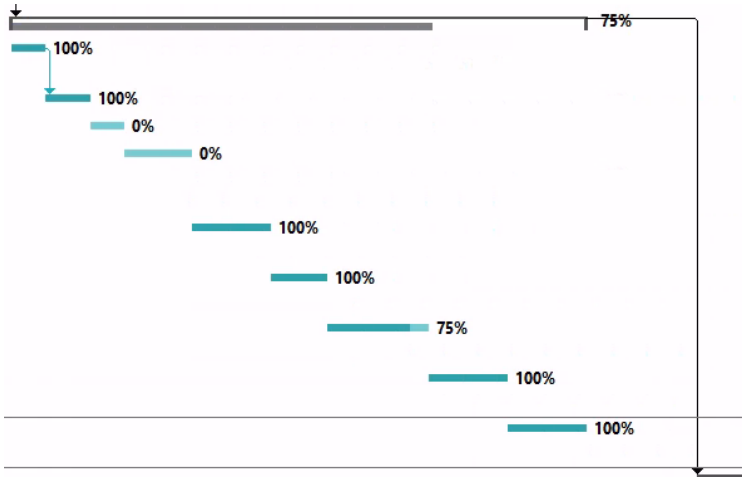
# 6 Project Plan



| | Task Mode | Task Name | Duration | Start | Finish | Predecess |
|---|---|---|---|---|---|---|
| 1 | | **Preliminary work** | **22 days** | **Mon 28/09/2** | **Tue 27/10/2(** | |
| 2 | | Idea brainstorming | 7 days | Mon 28/09/20 | Tue 06/10/20 | |
| 3 | | Research Ideas | 3 days | Wed 07/10/2 | Fri 09/10/20 | |
| 4 | | Lock in Idea | 2 days | Sat 10/10/20 | Mon 12/10/2 | |
| 5 | | Research similar products | 2 days | Tue 13/10/20 | Wed 14/10/20 | |
| 6 | | Figure out parts required | 2 days | Thu 15/10/20 | Fri 16/10/20 | |
| 7 | | Develop architecture diagram | 2 days | Sat 17/10/20 | Mon 19/10/20 | |
| 8 | | Draft proposal | 3 days | Tue 20/10/2( | Thu 22/10/2( | |
| 9 | | Getting supervisor advice on proposal | 1 day | Fri 23/10/20 | Fri 23/10/20 | |
| 10 | | Finish and submit final draft | 3 days | Sat 24/10/20 | Tue 27/10/20 | 2 |



| | | | Hardware side | 37 days | Wed 28/10/2 | Thu 17/12/2( | 1 |
|---|---|---|---|---|---|---|---|
| 12 | ✓ | | Draft up a parts list | 3 days | Wed 28/10/20 | Fri 30/10/20 | |
| 13 | ✓ | | Submit parts list | 3 days | Sat 31/10/20 | Tue 03/11/2( | 12 |
| 14 | | | Receive parts | 3 days | Wed 04/11/2 | Fri 06/11/20 | |
| 15 | | | Check boards with example code | 5 days | Sat 07/11/20 | Thu 12/11/20 | |
| 16 | ✓ | | Set up a wifi program | 5 days | Fri 13/11/20 | Thu 19/11/20 | |
| 17 | ✓ | | test the accelerometer | 3 days | Fri 20/11/20 | Tue 24/11/20 | |
| 18 | | | Figure out punch values | 7 days | Wed 25/11/20 | Thu 03/12/20 | |
| 19 | ✓ | | Push stats to webpage | 5 days | Fri 04/12/20 | Thu 10/12/20 | |
| 20 | ✓ | | Link with mongoDB | 5 days | Fri 11/12/20 | Thu 17/12/20 | |



| | | | Software side | 101 days? | Mon 28/12/2 | Mon 17/05/2 | 11 |
|---|---|---|---|---|---|---|---|
| 22 | ✓ | | Determine Software needed | 3 days | Mon 28/12/20 | Wed 30/12/20 | |
| 23 | ✓ | | Research Mean Stack | 3 days | Thu 31/12/20 | Mon 04/01/21 | |
| 24 | ✓ | | Set up sample project | 5 days | Tue 05/01/21 | Mon 11/01/21 | |
| 25 | ✓ | | Look into using mongoDB | 5 days | Tue 12/01/21 | Mon 18/01/21 | |
| 26 | ✓ | | Setup mongoDB | 5 days | Tue 19/01/2: | Mon 25/01/2 | |
| 27 | ✓ | | Enter db info into mongo | 5 days | Tue 26/01/21 | Mon 01/02/21 | |
| 28 | ✓ | | Push mongo data to node | 5 days | Tue 02/02/21 | Mon 08/02/21 | |
| 29 | ✓ | | Redesign Front End | 14 days | Tue 09/02/21 | Fri 26/02/21 | |
| 30 | ✓ | | Display Mongo stats on webpage | 7 days | Sat 27/02/21 | Mon 08/03/21 | |
| 31 | ✓ | | Develop a picker | 10 days | Tue 09/03/2: | Mon 22/03/2 | |
| 32 | ✓ | | Using the data create a chart | 7 days | Tue 23/03/21 | Wed 31/03/21 | |
| 33 | | | Implement next.js(optional) | | | | |
| 34 | ✓ | | Adding Search Database Feature | 10 days | Thu 01/04/21 | Wed 14/04/21 | |
| 35 | ✓ | | Add a Landing Page | 2 days | Thu 15/04/21 | Fri 16/04/21 | |
| 36 | ✓ | | Adding a mini navigation bar | 4 days | Sat 17/04/21 | Wed 21/04/21 | |
| 37 | ✓ | | Bug Fixing | 14 days | Thu 22/04/2: | Tue 11/05/2: | |

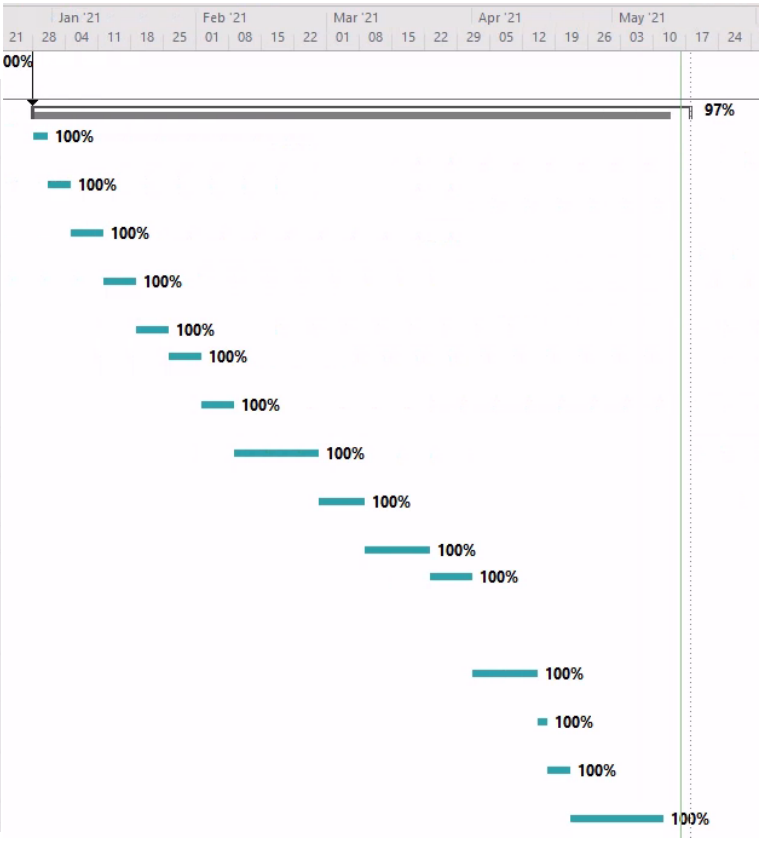



**Figure 6-1 Project Plan (Microsoft Project)**

# 7 React Native

In this section, I will discuss in detail the use of React Native and the reasoning behind using this approach. I will break down some of the key features that were used with regards to 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 my viewpoint. Creating native applications is a massive advantage that suited the style of the project I was making.

## 7.1 React Navigation

React Navigation is a convenient 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, and this is a standard container for all the screens needed. As shown below, we have created a navigation container, and then we declare our screen inside this. I have only used one screen in this example called “HomeScreen”.

```
<NavigationContainer>
  <Stack.Navigator mode="modal">
    <Stack.Screen
      name="Home"
      component={HomeScreen}
      options={{
        title: 'HomeScreen',
        headerStyle: {
          backgroundColor: '#708090',
        },
        headerTintColor: '#fff',
        headerTitleStyle: {
          fontWeight: 'bold',
        }
      }}
    />
```

**Figure 7-1 Stack Navigation Container**

The component name needs to be identical to the import component name to use a multi-file structure. As can be seen in “Figure 7-2”, the component that is being imported is identical in title to the element 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. 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.

```
import { HomeScreen } from './Screens/HomeScreen/HomeScreen';
```

**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 give an example and explain how this functions.

```
<FAB
    style={styles.fab3}
    large
    icon="information-outline"
    onPress={() => navigation.navigate('About')}
  />
```

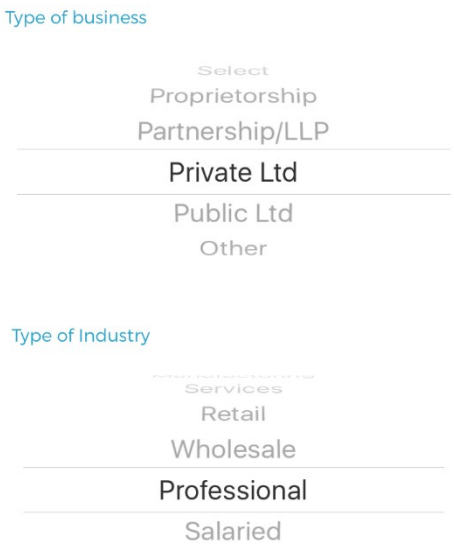
**Figure 7-3 Navigation Route**

When an action is performed, in this case, it’s “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 straightforward yet effective method of moving throughout various screens in react native applications

## 7.2 Picker

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

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

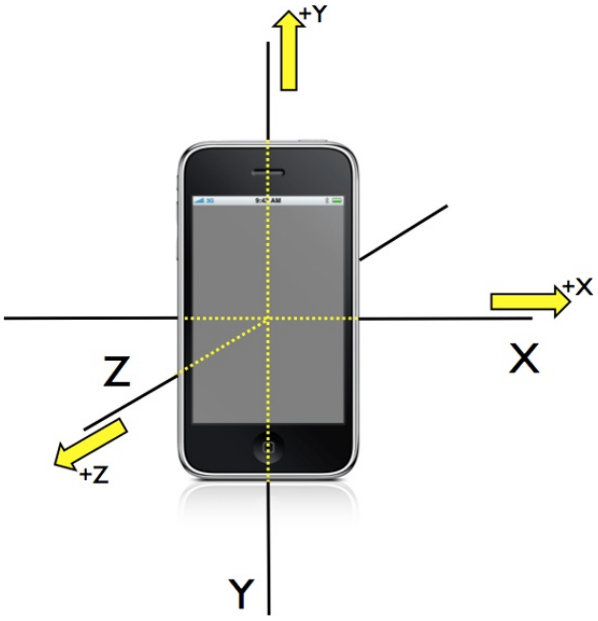



**[5]Figure 7-4 Picker Example (iPhone)**

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 that 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 later. I also used an additional picker when searching the database for previous bouts. When the name is searched, it fills the picker with the dates of the bouts retrieved from the “fetch API”.

# 8 Accelerometer

The Accelerometer is a crucial 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 concluded that harnessing the Accelerometer in my phone would be the quickest approach to solve the issue as time was lacking.

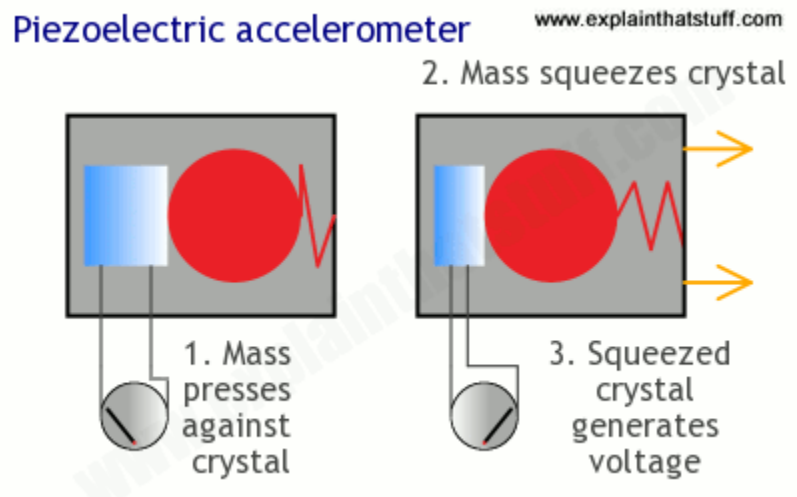



**[6]Figure 8-1 Accelerometer Axis**

As can be seen in "Figure 8-1", there is a 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 will increment the punch count. This is not the most accurate proposal; however, I was limited to this outcome due to a lack of time and resources.

## 8.1 How does it work?

"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 explained the concept of an accelerometer concisely.




**[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 severe 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, and this mass squeezes the crystal. The more mass supplied would indicate 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.

## 8.2 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 fundamental application that would harness the Accelerometer in a phone. It was not ideal; however, it does function and executes everything we need.

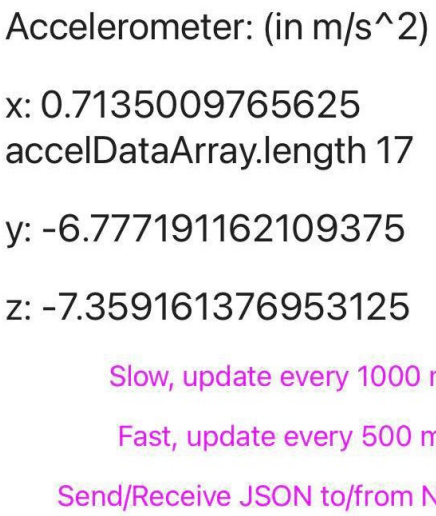



**Figure 8-3 Accelerometer Application**

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 stored and reset to zero when the round has ended, and the user has indicated that by pressing the stop round button.

```
router.post('/fetchPostExamaple', function(req, res, next) {
  console.log("received: ");
  console.log(req.body.firstParam);
  console.log(req.body.secondParam[1]);
  var newData = req.body.secondParam;
  for(var ii = 0; ii < newData.length; ii++){
    xData.push(newData[ii].x);
    if(newData[ii].x >0.5){
      x=x+1;
      console.log("punch thrown:"+ x);
    }
```

**Figure 8-4 Accelerometer Data Handling Code**

As shown 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 remove 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 reset for the next round.

# 9 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 offer an effortless analysis.

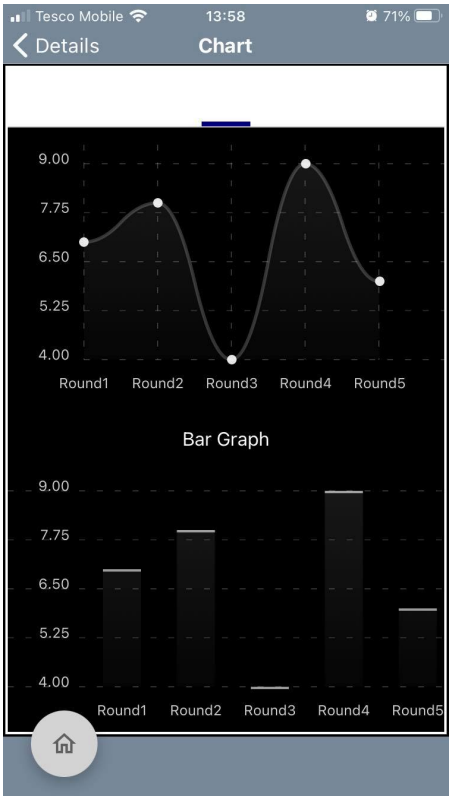
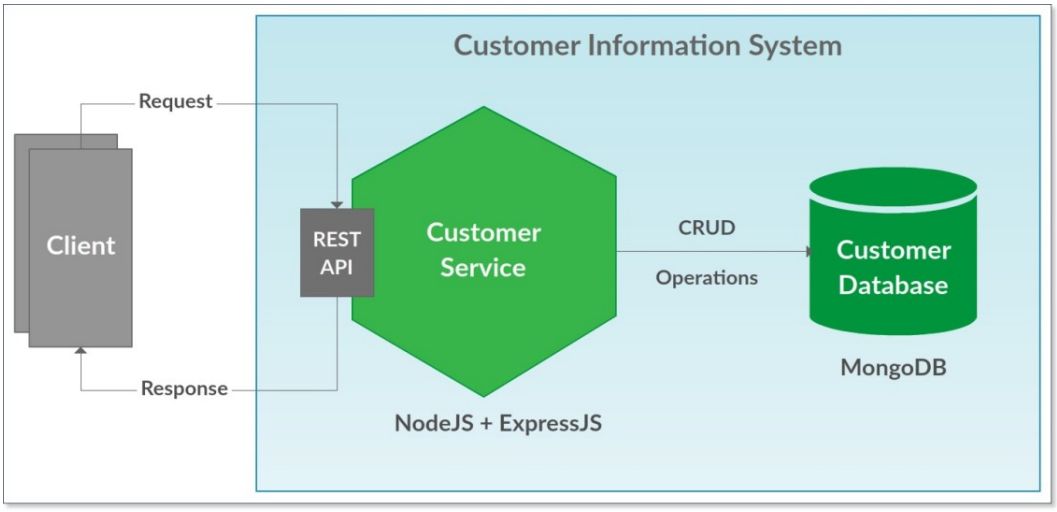



**Figure 9-1 React Native Chart Kit Example**

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 adequate and suited the style of data being gathered. The react native chart kit isn't as impressive on the web; however, it is incredibly professional on iPhone. Some of the charts available are Pie charts, Contribution graphs, and the two I have previously mentioned and used in my application.

# 10 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 a 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 10-1 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, and when a request is sent to the server, the appropriate response is sent back. For example, when a user ends the round, this sends a request to NodeJS. When NodeJS receives this request, it then handles it adequately and sends a response. When the user has stopped the round, the answer sent back from NodeJS to the user will be the punch total for that round.

NodeJS also communicates with MongoDB. MongoDB is my selected database, and NodeJS handles requests and responses from MongoDB also. You could almost say that NodeJS is the middle man or interpreter between my react native application and MongoDB database.

# 11 Expo

“Expo is a toolchain built around React Native to help you quickly start an app. It provides a set of tools that simplify the development and testing of React Native app and arms you with the components of users interface and services that are usually available in third-party native React Native components”[11]. Expo provided me with an excellent platform to develop my react native application. There are great libraries and components that the user can avail of.


**[12]Figure 11-1 Expo Logo**

Expo provided me with tremendous documentation. It gives detailed tutorials showing developers how to integrate both components and libraries into projects. I was fortunate to have access to Expo as I felt it had a smooth learning curve. The step by step tutorials provided me great support when I was running into difficulty and debugging my code. It provides an ideal platform for anyone learning how to code cross-platform applications with react native.

# 12 MongoDB

MongoDB was the most logical choice following the MERN stack approach. MongoDB provided me with all the core functionality required, and it worked seamlessly with NodeJS.

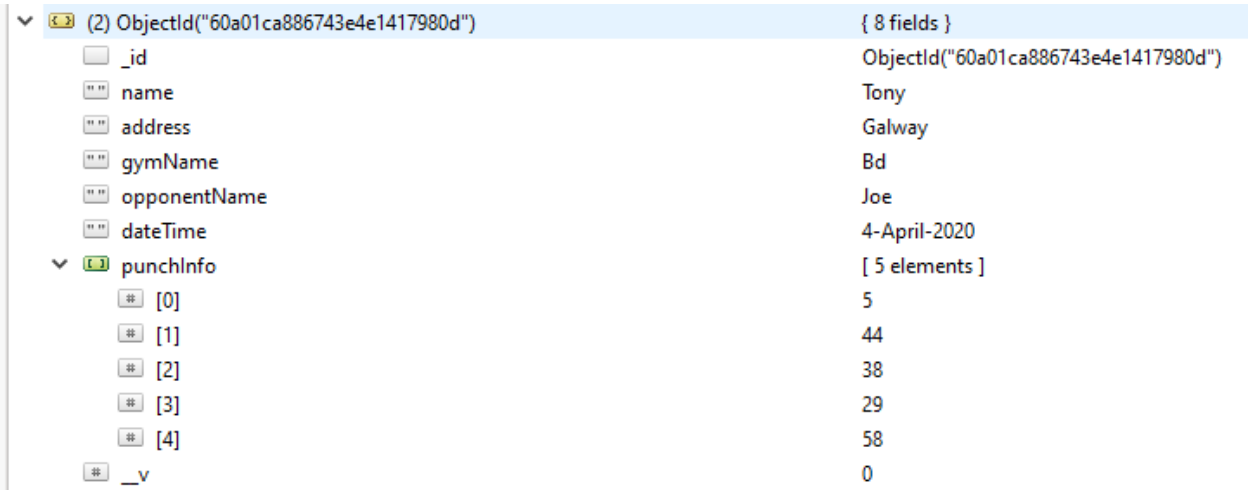



**Figure 12-1 Database Example**

In “Figure 12-1”, we can see the data structure of the objects stored in my database. I created a collection in MongoDB that holds these objects. Each object contains all the essential details for a specific bout, and as can be seen, an array of length five to store the five rounds of punch totals. I found the database side challenging initially; however, once you can master the basic commands, it becomes much more effortless.

When a search is taking place, the user will pass the name of the fighter. NodeJS will pull all the objects from MongoDB with the specific name. The application then fills a picker (dropdown list) component with the dates of the objects. The user can then choose by the bout’s date to gain access to the specific bout in question.

# 13 Research and Planning

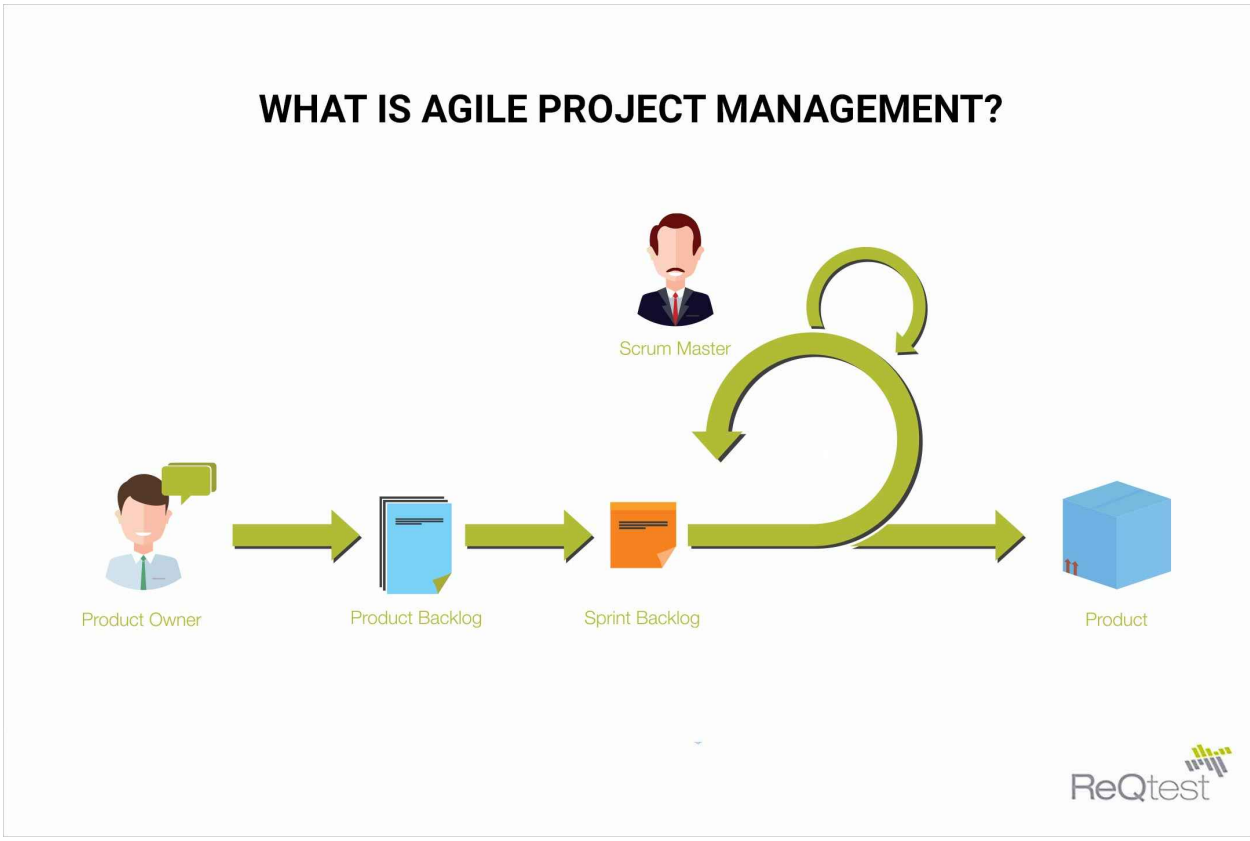
## 13.1 Research

Research plays an essential role in any well-planned project. Once I had a general idea of what I wanted to create, I searched for information to assist me. Firstly I explored the internet looking for similar applications to my proposal. I wanted to discover if any application existed identical to my idea. I did stumble upon a similar product; however, it was more aimed towards beginners and teaching them basic boxing classes and punch fundamentals. This application did also gather punch data, and it completed this by a specially crafted hand wrap that is paired with the application on the user's phone.

As I had very few projects similar to my idea, it took a bit of experimenting and a lot of research to decide whether I could turn this proposal into a working project. Carefully I gathered data on accelerometers and chose the best approach was to use an "Esp 32" [13] and an accelerometer circuit board. I also researched the MERN stack and how this would be beneficial to my project. Overall, research played an essential part in this project, and without solid research, this project would not have been possible.

## 13.2 Planning

Once I had researched the idea and realised how this could be completed, I planned this project. I utilised the tool "Microsoft Project" for the planning of this project. This tool is an excellent project management program that allows the user to create tasks and then create smaller, more achievable jobs called "Sprints" [14]. This is a form of agile project management where the team or induvial breaks down a large project into smaller, clearer tasks. The group or individual can work towards these small short term goals, which adds up to one big job completed.




**[15]Figure 13-1 Agile Project Management Diagram**

This approach was motivating as once all the small tasks come together, the more significant task or goal doesn't seem as too big of a mountain to climb. I have attached a diagram above showing the agile project management approach in a professional environment. This is a very efficient manner of completing a project. I was fortunate to be exposed to this project management style throughout my work placement, and it proved to help my approach in this specific project.

# 14 Ethics

Ethics are an essential part of software engineering and have been nested throughout my project. Some things to consider are data protection, intellectual protection, and providing references where code and components were taken.

I was fortunate to have the help of my supervisor to help guide me down the right path of this project. Credit is commented in the code where it is appropriate. Without this credit, it is difficult to determine who coded specific parts. It is also an ethical thing to do, and credit individuals work when appropriate as we are lucky to have access to other developer's code.

Data protection is vital in today's world. In this project, I don't have any sensitive data. However, I am still using 'POST' rather than 'GET' as 'POST' is much more secure than a 'GET' request. Even though it's not required here, I still feel it's the honest approach to develop the application more securely.

Intellectual property ties in with my first point. It's essential to credit others intellectual property, and it is not ethical to claim others ideas and work as our own.

# 15 Conclusion

Briefly, to conclude, I would like to share the overall outcome of this project. I was content with the final result. The project I envisioned at the beginning of the academic year is mostly what I have created now. Originally I wanted to detect punches in different directions; however, I didn't achieve this, and I decided to focus more on seeing a single punch first. I felt that it was the best starting block and enabled me to present the core functionality of my project. The general idea of the project functions correctly. The application gathers real-time data and stores this data in a database for future reference. Overall, I am delighted with the outcome of this project and the skills I have obtained over this module. I was granted a valuable opportunity to absorb as much knowledge about full-stack development as possible, and I feel that my skills have been enhanced.

Regarding the future development of this application, I would hope to implement punch counters in their relative directions, such as hooks and uppercuts. Also, I would love to explore the hardware aspect in greater detail and produce an adequate solution to place an accelerometer in a glove or hand wrap.

# 16 Appendix

**Abbreviations**

AWS – Amazon Web Services

FAB – Floating Action Button

API – Application Programming Interface

JSON – JavaScript Object Notation

DB – Database

MERN – MongoDB-Express-React-NodeJS

**Miscellaneous**

https://github.com/oisinconcannon/projectlog

**Contact Information**

Oisin Concannon,

G00347603@gmit.ie ,

GMIT,

Galway

# 17 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

[11] Expo vs Vanilla React Native: https://apiko.com/blog/expo-vs-vanilla-react-native/#:~:text=Expo%20is%20a%20toolchain%20built,party%20native%20React%20Native%20components.

[12] Expo Logo Image: https://static.expo.dev/static/brand/all-logos.png

[13] Esp 32: https://en.wikipedia.org/wiki/ESP32

[14] Everything you need to know about sprints by medium.com: https://medium.com/@concisesoftware/everything-you-need-to-know-about-sprints-in-project-management-4b378a7eb83f#:~:text=A%20Sprint%20is%20a%20short,software%2C%20faster%20and%20more%20frequently.

[15] Project Management Diagram Image: https://reqtest.com/wp-content/uploads/2020/05/Agile.jpg