



TUS
Midwest

Department of
Information Technology

An Investigation of Data Analytics and Modern Technologies and
Their Potential Use in a Sports Club Management Application

By John Power K00250447

A Final Year Project submitted in partial fulfilment of the
requirements of Technological University of the Shannon for the
degree of Bachelor of Science (Honours) in Software
Development

Supervised by: Brendan Watson

Ethical Declaration

BSc in Software Development – Declaration of Authorship

I, John Power, declare that the software development project titled “An Investigation Of Data Analytics and Modern Technologies and Their Potential Use in a Sports Club Management Application” submitted by me as part of my BSc in Software Development is entirely my own work, and I have not used unauthorized external assistance in its creation. I affirm that:

- I have fully and independently developed the software code, design, and documentation contained within this project.
- Any external sources, such as code snippets, libraries, or third-party resources, have been appropriately cited and referenced in accordance with academic citation standards.
- This project is a result of my own creative and intellectual efforts, and I have not copied, borrowed, or utilized work produced by others without proper acknowledgment.
- I have not used automated content generation tools, including AI-based code or text generation services, in the creation of this project.
- Any collaborations or contributions from external individuals are acknowledged and appropriately credited in the project documentation.
- The code and documentation presented here reflect my understanding of the concepts and principles learned during the course.

I understand that any breach of academic integrity, including plagiarism or submitting work that is not my own, is subject to disciplinary actions as outlined in the TUS academic policies.

[Signature]

John Power

[Date]

_17/03/2025

Abstract

The purpose of this project is to research into how data analytics and modern technologies are used in sports. The author then developed a sports club management web application that applies descriptive analysis and the potential use of a GPS device to record team session statistics. The research chapters consist of comparing sport applications used by sport clubs, the effect of modern technology in sport, data analytics in sports, how software developers incorporate data analytics into software, data visualization techniques and web application frameworks and the features they have to offer. The author also describes the design approach, the technology that was utilised and how the author implemented the web application while also discussing the challenges he encountered and future workings of the project.

Acknowledgements

The author would like to thank his project supervisor Brendan Watson for his tremendous support and guidance throughout the research and development process of this project.

The author would also like to thank his parents for their continuous love and support throughout the authors past four years at the Technological University of The Shannon.

The author would also like to thank all of the lecturers at the Technological University of The Shannon for sharing their skills and knowledge to the author for the past four years which helped him to carry out this project.

Table Of Contents

1	Introduction	1
2	Literature Review	2
2.1	Sport Applications Used in Sport Clubs	2
2.1.1	ClubZap	2
2.1.2	Teamer	2
2.1.3	Hudl	3
2.1.4	Comparing ClubZap, Teamer and Hudl	3
2.2	The Effect of Modern Technology in Sport	4
2.2.1	Wearable Technologies	4
2.2.2	Artificial Intelligence	5
2.2.3	Virtual Reality	5
2.2.4	The Effect Of Modern Technologies In Sports Chapter Conclusion	6
2.3	Data Analytics in Sports	6
2.3.1	The Impact of Data Analytics in Sports	6
2.3.2	The Obstacles of Data Analytics in Sports	8
2.3.3	Data Analytics in Sports Chapter Conclusion	8
2.4	How Do Software Developers Incorporate Data Analytics Into Software?	8
2.4.1	Data Logging	9
2.4.2	Embedded Analytics	10
2.4.3	Predictive Analytics	13
2.4.4	How Do Software Developers Incorporate Data Analytics Into Software Chapter Conclusion	15
2.5	Data Visualization	15
2.5.1	Pie Charts	15
2.5.2	Line Charts	16
2.5.3	Scatter Plot Graphs	17
2.5.4	Application Of Data Visualization in Sports	18
2.5.5	Data Visualization Chapter Conclusion	19

2.6	Web Application Frameworks	19
2.6.1	Spring Boot.....	20
2.6.2	Django.....	20
2.6.3	Angular	21
3	Design & Analysis	23
3.1	Functional Requirements.....	23
3.2	Non-Functional Requirements.....	24
3.3	Agile Development	24
3.4	Use Cases.....	25
3.4.1	Club Members Use Cases	25
3.4.2	Club Player Use Cases	26
3.4.3	Club Trainer Use Cases	27
3.4.4	Club Administrator Use Cases.....	29
3.5	Technology Stack Utilised in the Project.....	31
3.6	Sports Club Management Web Application Components	33
3.7	System Architecture.....	34
3.8	Database Design	35
4	Implementation.....	37
4.1	Spring Boot Security.....	37
4.2	Utilising Thymeleaf Fragments	39
4.3	User JPA Entity Example.....	41
4.4	Registering A Member ThymeLeaf Form	44
4.5	User Controller Layer Method Example.....	46
4.6	Create User Service Layer Method	47
4.7	Designing the Navigation Bar and Tables Using Bootstrap.....	48
4.8	Utilising AJAX For Editing Team Details	49
4.9	Implementing Graphs Using D3.....	50
4.10	Uploading Team Session Statistics CSV File	60
4.11	REST API To Export Team Session Statistics to a PDF File.....	62
5	Testing And Results	65
5.1	Functional Testing.....	65
5.2	Test Cases	65

5.2.1	Add A Club Event	65
5.2.2	Log Club Event Attendance	66
5.2.3	Add A Team.....	67
5.2.4	Edit A Team.....	69
5.2.5	Add A Team Session	70
5.2.6	Upload Team Session Statistics CSV File	71
5.2.7	Add Player Physical Stats	72
5.2.8	Export Team Session Statistics To A PDF File	73
5.2.9	Members Distribution By Gender Bar Chart.....	74
5.2.10	Player Physical Stats Line Chart.....	76
6	Challenges Encountered in The Project.....	79
7	Future Work.....	81
8	Conclusion	83
9	References	85

List Of Figures

Figure 1: An Example of A Python Pie Chart To Display The Popularity of Each Sport (Alex, 2021).....	16
Figure 2: A Simple Line Graph Displaying The Number of Bicycles in the First 6 Months of the Year (CueMath, no date)	17
Figure 3: An example of a scatter plot being used to display the correlation between the number of matches played and the age of players (Thrane, 2023).....	17
Figure 4: Power BI dashboard created by Miami Heat (Briggs, 2019).....	18
Figure 5: Club Members Use Case Diagram	25
Figure 6: Club Player Use Case Diagram	27
Figure 7: Club Trainer Use Case Diagram	29
Figure 8: Club Administrator Use Case Diagram.....	30
Figure 9: Sports Club Management Application Component Diagram	33
Figure 10: Sports Club System Architecture Diagram	34
Figure 11: Sport Club DB database design.....	35
Figure 12: Load User By Username Method.	38
Figure 13: Fragments file structure.....	40
Figure 14: Fragment tag for navigation bar.	40
Figure 15: Insert tag to insert navigation-header fragment.....	40
Figure 16: User Entity JPA Class.....	42
Figure 17: Register A Member Thymeleaf Form	44
Figure 18: Register Member DTO Form Class	45
Figure 19: Add Club Member Controller Method.....	46
Figure 20: Create User Service Method	47
Figure 21: User Repository Interface	48
Figure 22: Navigation Bar Bootstrap Classes.....	48
Figure 23: Log Out Button Design using Bootstrap	48
Figure 24: Navigation Bar	49
Figure 25: Table Bootstrap Classes	49
Figure 26: Club Events Table Example.....	49
Figure 27: Add Team Member and Remove Team Member Functions using AJAX	50

Figure 28: Converting a List Of Maps to JSON String using the Jackson Library	51
Figure 29: Output of Club Admin Gender Count List in JSON string Format	51
Figure 30: JavaScript Functions Being Called To Draw Graphs.	52
Figure 31: Draw Gender Chart Function	53
Figure 32: Buttons That Update the Bar Chart Depending on the Role Selected.	54
Figure 33: Update Gender Chart Data Function	55
Figure 34: Update Gender Chart Function	56
Figure 35: Members Distribution Bar Chart Displaying the Number of Male and Female Members	57
Figure 36: Draw Player Physical Stats Line Chart Function	58
Figure 37:Update Line Chart Function	59
Figure 38: Line Chart Showing Jack Keogh's Weight Trends Over Time	60
Figure 39:Upload Team Session Stats CSV File HTML Form Code	61
Figure 40: Create Team Session Stats From CSV File Controller Method	61
Figure 41: Create Team Session Stats For Team Session From CSV File Service Method	62
Figure 42: Team Session Stats PDF REST Controller Method	63
Figure 43: Generate Team Session Stats PDF Service Method	64
Figure 44: Add Club Event Test Data	66
Figure 45: Add A Club Event Output	66
Figure 46: The Log Event Attendance Output	67
Figure 47: John Doe's Attendance Displayed on The Hurling Final Club Event	67
Figure 48: Add A Team Test Data	68
Figure 49: Add A Team Output	68
Figure 50: The New Senior Hurling Team Details.	68
Figure 51: Edit Team Test Data	69
Figure 52: The Edit Team Output	70
Figure 53: The Senior Hurling Team Updated Details.	70
Figure 54: The Add Team Session Test Data.	71
Figure 55: The Add Team Session Output.	71
Figure 56: Hurling Training Team Session Stats CSV File.	72
Figure 57: The New Team Session Statistics Details.	72

Figure 58: Add Player Physical Stats Test Input.....	73
Figure 59: Add Player Physical Stats Output.....	73
Figure 60: Hurling Training Recorded Team Session Statistics.	74
Figure 61: Generated PDF File Containing The Hurling Training Team Session Statistics	74
Figure 62: Bar chart displaying the number of Male and Female Members.	75
Figure 63: Bar Chart Displaying the Number of Male and Female Players	76
Figure 64: Jack Keogh's Recorded Physical Stats	77
Figure 65: Line Chart displaying Jack Keogh's Weight Trends Over Time.....	77
Figure 66: Line Chart Displaying Jack Keogh's Body Fat Percentage Trends Over Time.....	78

List Of Tables

Table 1: Features provided from ClubZap, Teamer and Hudl Applications3

List Of Abbreviations

AI	Artificial Intelligence
AJAX	Asynchronous JavaScript and XML
AWS	Amazon Web Services
API	Application Programming Interface
BMI	Body Mass Index
BPM	Beats Per Minute
Bsc	Bachelor of Science
CEO	Chief Executive Officer
CSV	Comma Separated Values
D3	Data-Driven Documents
GAA	Gaelic Athletic Association
GPS	Global Positioning System
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IOC	International Olympic Committee
IRFU	Irish Rugby Football Union
JPA	Java Persistence API
JSON	JavaScript Object Notation
JSP	JavaServer Pages
KAGR	Kraft Analytics Group
KPI	Key Performance Indicator
MLB	Major League Baseball
MVC	Model View Controller
MVT	Model View Template
NBA	National Baseball Association
NFL	National Football League
ORM	Object Relational Mapping
PDF	Portable Document Format

REST	Representational State Transfer
SPA	Single Page Application
SQL	Structured Query Language
TUS	Technological University Of The Shannon
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
VR	Virtual Reality
XAMPP	Cross-Platform, Apache, MariaDB, PHP and Perl

1 Introduction

Throughout his life, the author had a strong interest in sports. Data analytics and the use of modern technologies has been making a big impact in sports in recent years. The author wanted to integrate the skills he learned throughout his past four years in TUS with his love of sports by developing a sports club management application that applies descriptive analytics and the potential use of GPS devices to record team session data.

The purpose of this project was to research into how data analytics and modern technologies are used in sports and then develop a sports club management web application that generates statistics on members, players, team sessions, club events. These statistics are then displayed to the user using suitable data visualisation techniques such as tables, bar charts and line graphs to help them easily identify patterns or trends in the sport club's data that could help them make informative decisions to improve the sports club. Team session statistics could potentially be recorded by each player wearing GPS tracking devices in which the data can then be exported to a CSV file that can be recorded to the sports club system.

Before developing the application, The author researched into topics such as applications used by sports club, the effect of modern technology in sports, data analytics in sports, how software developers incorporate data analytics into software, data visualization techniques, and also web application frameworks that could be utilized. This research helped the author in defining the web applications requirements that it should provide.

The author talks about the design approach and the technologies used to implement the web application. The author will explain how he implemented important pieces of functionality of the web application by including snapshots of code the author wrote. The author also outlined suitable test cases he used in order to test key features of the application and also documented their results.

The author discusses about the challenges he encountered in the project and the areas that need to be improved in future iterations of the project. He also gives his overall opinion on the work he completed in this project.

2 Literature Review

In this chapter, the author discusses about the research he carried out in topics such as sport applications used in sport clubs, the effect of modern technologies in sports, data analytics in sports, how software developers incorporate data analytics into software, data visualization and web application frameworks.

2.1 Sport Applications Used in Sport Clubs

The author has used various applications throughout his life in order to get sport results, news and also as a form of communication for upcoming training sessions or matches. The author researched into three sport applications that are used by sport clubs to get insights into what type of different services these applications have to offer. These three applications consisted of ClubZap, Teamer and Hudl. The author is already familiar with ClubZap as he uses the application to keep informed of news, fixtures and results from his local GAA club. The author researched a blog post by Practice which listed the ten best sport applications based on different categories in which the author focused on Teamer and Hudl (Practice, 2023).

2.1.1 ClubZap

ClubZap is a mobile and web application that enables sports clubs to communicate with their club members. The application can provide membership management with payment collection for sport clubs which makes it easier to manage complex club members data while ensuring that payments are stored in the club's bank account within a week of payment. The payment collection feature on the ClubZap application allows for sport clubs to organise fundraising activities in which they can collect donations from club members and also be able to communicate with club members to explain what their money is going to be used for. It's essential to have a strong communication outlet with club members, that is why they provide a chat service to allow members to send messages to the club and also enables integration with your club social media accounts. Club news, fixtures and results can be automatically published to your social media accounts making it easier for club members to keep up to date. (ClubZap, no date)

2.1.2 Teamer

The Teamer application provides a platform that enables users to efficiently manage their sports team. The services that the Teamer application can provide includes a messaging service that allows for easy communication with other team members, organise events

such as training sessions, matches and other team events in which invites can be sent to team members so they can log their attendance. Teamer utilises Apple and Android pay services to enable users to easily collect payments and manage transactions, it also notifies members who haven't paid up their fees. (Teamer, no date)

2.1.3 Hudl

Hudl provide data analytics to sport teams and organizations around the world through the use of their cameras, wearable devices and software. Hudl provide Focus cameras which is capable of recording hands-free footage and livestreaming which sports teams can utilise to record their games or training sessions to generate key analytics and statistics from every player on the team using the Hudl platform. They provide their services to more than 230,000 teams from more than forty sports (Hudl, no date)

2.1.4 Comparing ClubZap, Teamer and Hudl.

	ClubZap	Teamer	Hudl
Features:	<ul style="list-style-type: none"> • Club Membership management service • Collect Payments • Chat Service • Manage Club News, Events, Fixture and Results 	<ul style="list-style-type: none"> • Team management service • Payment Service • Organise team events and invite team members. • Log attendance to events. 	<ul style="list-style-type: none"> • Sport Data analytics service • Provide cameras for recording matches and training sessions • Provide wearable devices

Table 1: Features provided from ClubZap, Teamer and Hudl Applications

In Table 1, each application provide services that are beneficial to sport clubs. The author uses the ClubZap app which is very useful for keeping him informed about club news,

upcoming club events, upcoming fixtures and match results and also have the option to pay for club membership or purchase lotto number lines for the club lotto. Teamer is more focused on managing sport teams and also offer services such as payment collection, organising team events for team members in which they can log attendance (Teamer, no date). The ClubZap and Teamer applications are similar in which they provide similar features to help users to manage their sports club or team. Hudl has more of a focus on helping sport clubs gathering analytics through their cameras that can record team sessions or matches and also provide their software platform to generate insightful analytics from the recorded footage to help improve the performance of sports teams.

2.2 The Effect of Modern Technology in Sport

Technology is widely used in society today and it especially has a significant role in the sports sector. The author researched an article from Team EMB about how certain technologies in sport are improving performance and experience. The author decided to do more research on how wearable technologies, artificial intelligence and virtual reality is making its effect in the sports industry (Team EMB, 2024).

2.2.1 Wearable Technologies

Wearable technologies such as GPS devices and impact monitor stickers play a beneficial role in improving athletes performance and how they train. Athletes GPS tracking devices in order to record metrics such as heart rate, speed and distance covered using real-time data that helps athletes and coaches to improve their training plans (Team EMB, 2024). Not only does wearable technology help improve an athlete's performance in the field, but it can also detect early signs of tissue damage in muscles which allows athletes to change their training plans to prevent further injury with the use of impact monitor stickers which can also aid in detecting potential head injuries such as concussions (Cogniteq, 2023).

An example of a GPS device that is widely used by professional sports clubs such as Arsenal Football Club, USA Womens National Football Team, IRFU and the Washington Commanders is the Apex GPS device developed by Stat Sports (Stat Sports, no date a). Stat Sports offers their Sonra Lite service that provides two solutions which consist of the coach led or player led. The coach led solution allows coaches to manage the Apex GPS devices for their team and sync the recorded data to the Sonra Lite web application while the player led solution allows players to download their own sessions that is

recorded from their Apex GPS device to a mobile application which enables players to analyse their own session. Their coach led solution records up to 18 metrics such as distance covered, max heart rate and are also able to export the data recorded from the devices to a CSV file (Stat Sports, no date b).

2.2.2 Artificial Intelligence

AI also has a significant role in the sports industry, AI also helps with preventing further injury through the use of machine learning and data analytics to detect patterns that could lead to injuries due to overtraining, this allows trainers to adjust the athlete's training plan. Sport analytics has become an easier process as AI is able to analyse large amounts of data in a shorter span of time to allow coaches to make decisions during matches and help in predicting possible match outcomes enabling coaches to adjust their game plan (Team EMB, 2024).

In the 2024 Olympics, the IOC used AI to help protect athletes from cyber abuse by being able to monitor social media accounts to flag abusive messages towards Olympic athletes. According to the IOC president Thomas Bach, they were expecting an estimated half a billion social media posts throughout the 2024 Olympic games (International Olympic Committee, 2024).

The use of AI chatbots and virtual assistants is allowing sport organizations to interact with consumers. These chatbots are able to answer questions from fans, send real-time updates and can recommend content to fans or even ticket-options. AI chatbots can help with improving fan loyalty and engagement through providing a more immersive fan experience. (Team EMB, 2024).

2.2.3 Virtual Reality

VR in sports has provided sport teams the opportunity to practice real-world scenarios that could occur in games(Team EMB, 2024). According to Lissa Mazza, she mentions that professional teams such as the Dallas Cowboys, the Philadelphia 76ers and the Tampa Bay Rays have utilised VR technology in order to train in a non-contact environment that helps in preventing unnecessary injuries and overtraining (Mazza, 2018).

The use of VR technology can provide many benefits to athletes and coaches but there are certain factors such as equipment and technology development which is very

expensive. It's also essential that VR Simulations are of high quality and accurate (Team EMB, 2024).

2.2.4 The Effect Of Modern Technologies In Sports Chapter Conclusion

Wearable technologies such as GPS tracking devices allow athletes to record metrics such as their heart rate, speed and distance covered using real-time data that can allow them to improve their training sessions and the use of impact monitor stickers can help athletes and trainers to detect potential injuries. After researching into the Apex GPS device, the author decided to apply for a demo of the Apex GPS Device since it is possible that the recorded data can be exported to a CSV file, the author didn't get a reply from Stat Sports but decided to implement a feature that allows trainers to upload team session statistics recorded on a CSV file as a proof of concept. AI technologies can help with preventing further injury, help coaches with better decision making and adjusting their game plan and help provide an immersive fan experience through the use of AI chatbots. It was also used by the IOC to protect Olympic athletes from cyber abuse during the 2024 Paris Olympic Games. VR technologies have been utilised by the Dallas Cowboys, Philadelphia 76ers and the Tampa Bay Rays to train in a non-contact environment to prevent injuries and overtraining. The author believes that the use of technology in sports is making a positive impact to sports organisations and will be intriguing to see what effect it is going to have in the future. Although the application developed by the author doesn't consist of AI and VR technologies, they are still relevant modern technologies that are used in sport.

2.3 Data Analytics in Sports

Data has always been essential to the growth of businesses in any industry especially in sports. It was in the late 20th century that sport analytics was introduced but it would be early in the 21st century that sport analytics would play a bigger role due to the advancements of technology and digitalization(Catapult, 2024). The author decided to research more into what the impact of data analytics has in sports and what obstacles are faced when trying to use sport analytics.

2.3.1 The Impact of Data Analytics in Sports

The use of data analytics in sports is providing many benefits including better decision making, helping teams to achieve better results and improving fan engagement. With the

surplus amount of data being generated and it being more accessible to the public, many people involved in the sports industry are applying analytics in order to gain an edge amongst their competitors(Pykes, 2022).

Improved Decision Making

Data analytics can help sport teams to enhance their decision making that can provide a positive impact to the team's performance on and off the field (Pykes, 2022). According to Kurtis Pykes, mulitiple NBA teams inculding the Philidelphia 76ers have utilised data visualization and hypothesis testing to analyze NBA games which influenced their coaching strategy (Pykes, 2022).

Helping Teams to Achieve Better Results

Sport analytics is being used to help improve the overall performance of teams which can also help in producing better results. A good example of this is the Oakland A's 2002 season in which Billy Beane implemented data analytics to create a winning baseball team on a limited budget (Adam, 2022). Beane and his assistant Paul DePodesta focused on players that had statistics such as high on-base percentages which correlated with run-production and winning games. Players with high-on-base percentage were usually cheaper to acquire as they didn't seem to be valuable players but played a valuable role for the Oakland A's as they would win 20 games in a row and also win the American League West Division (Goldstein, 2017)

Improving Fan Engagement

The application of data analytics has helped to increase fan engagement, sports organisations to discover more about their fans behaviours by monitoring what the fans are watching online through online video views, they also gather data from fans entering stadiums through the use of electronic tickets, fingerprint and retinal scans (Ricky, 2019). In a 2016 article for Wbur, Shira Springer talks about how the New England Patriots worked with KAGR to improve the fan experience. KAGR monitored fan behaviour from Pro Shop purchases, when game tickets were purchased, whether team emails were actually being read, when fans would start showing up at the stadium's parking lot and also how many ticket holders would attend the game. They were able to retrieve detailed information on fans who purchased tickets and didn't attend the game. The president of the New England Patriots at the time, Johnathan Kraft, found this data helpful to find out who is missing these games and the reasons why. The CEO of KAGR Jessica Gelman

believed that the Patriots could make more informed decisions in areas such as the price of tickets, game-day staff and stadium improvements through gathering as much data as possible (Springer, 2016).

2.3.2 The Obstacles of Data Analytics in Sports

Although the use of data analytics can provide many benefits to sports organisations on and off the field, there is some challenges in implementing sport analytics that needs to be noted.

- **Gathering Quality Data:** There are large amounts of data available for sports organisations to use but it can be an issue to try and clean up these vast amounts of data. There is a possibility that some values may not be recorded or inaccurate which could lead to inaccurate results(Data Overload, 2023).
- **Human Element:** There are times where sport analytics doesn't consider player motivation and injuries which is challenging to analyse. These are human elements of the game which could determine the finishing result of games. It is also difficult to persuade coaches and players to use analytics to select their strategies or make decisions (Data Overload, 2023).
- **Privacy Protection:** Although the use of sport analytics can provide many benefits into an athlete's performance, it is important that their private information is secure. It can be challenging to protect athletes' private data from being leaked. It is essential that there is strong authority control and implement data desensitization methods to protect athletes' private data (Bai and Bai, 2021).

2.3.3 Data Analytics in Sports Chapter Conclusion

Data analytics in sports is making an impact by improving decision making, enabling teams to achieve better results and also improving fan engagement. Although the use of sport analytics can provide many benefits, there are factors such as gathering clean data, human elements and privacy protections which need to be considered.

2.4 How Do Software Developers Incorporate Data Analytics Into Software?

Data Analytics play a crucial role in various industries, including software. Software developers can utilise data analytics through different approaches depending on the purpose of the application and the type of data that is going to be analysed (Wright, 2023). The author researched more into three of the many ways in which data analytics can be

utilised, Theses three examples included Data Logging, Embedded Analytics and Machine and Predictive Analytics

2.4.1 Data Logging

Logging plays an important role in Software Development as it helps software developers to build and maintain a smooth-running application for users. Developers use data logging tools to enable them to record data from their software system such as events, activities and log messages produced from the system. Developers can analyse this log data to help identify bugs, application behaviour, monitoring performance and be able to track the behaviour of users. Without data logging, the software development process would be more complex and time consuming.(Wright, 2023)

Purposes Of Data Logging

Data Logging has multiple purposes in the Software Development process. Some of these purposes include detecting and fixing bugs, traceability and troubleshooting and enhancing user experience.(Wright, 2023)

Detecting And Fixing Bugs

When an error occurs in an application, a log entry is generated which provides details such as the systems state or exceptions that occurred. This information can be used by developers to discover what exactly happened to the system and be able to refactor the code to resolve the error (Wright, 2023).

Traceability And Troubleshooting

Data logging allows developers to implement traceability and troubleshooting on their systems which is one of the main purposes of logging. When an application encounters an error, developers are able to analyse the log files to discover where exactly the error occurred in their code. This information makes it easier to locate the error and fix it.(Wright, 2023)

Enhancing User Experience

Developers are able to record insights on user interactions with their application which can help to identify user behaviour patterns and their preferences. This data can be used to improve the user experience of the application through making informed UI/UX decisions and also focus on feature development (Wright, 2023).

These are just few examples of the many purposes that data logging provides to software developers.

What Do Developers Use to Implement Data Logging into Software

There are multiple tools and frameworks that are utilized by developers which helps to simplify the process of implementing and managing logs. These include Log4j and Winston.(Wright, 2023)

Log4j

Log4j is a popular logging library used in Java. It consists of an API and multiple components such as appenders, layouts, filters and lookups.

- **Appenders:** Their appenders are able to deliver log events to various destinations such as files, sockets, databases, etc (Log4j, no date).
- **Layouts:** Layouts are able to generate CSV, HTML, JSON formatted outputs.
- **Filters:** filters can be applied to log events based on regular expressions, log event rates, time, etc (Log4j, no date).
- **Lookups:** Allow developers to gain access to system properties, environment variables, log event fields, etc (Log4j, no date).

Log4J also supports performance, reliability and extensibility for developers . (Log4j, no date)

Winston

Winston is a widely used logging library for Node.js. It decouples different features such as log levels, formatting and storage in order to make logging more extensible and flexible. It also helps to make API's independent and support multiple combinations. The Winston also utilizes Node.js streams which helps to reduce performance impact due to incorporating logging into applications.(Ayooluwa, 2025)

2.4.2 Embedded Analytics

Embedded Analytics consists of integrating data analysis and visualization capabilities into business applications, portals or websites to enable users to analyse data on the platform instead of third-party applications. Embedded Analytics allows businesses to easily interact with data by making it more accessible, usable and interpretable for all users to help them enhance productivity, user experience and efficiency through making

data driven decisions (Datafeer, no date). Some of the key features of embedded analytics platforms includes dashboards, data visualization, self-service analytics, data connectors and interactivity (Mostert, 2023).

Benefits Of Embedded Analytics

The ability for users to easily and efficiently access data on the same interface without having to go to an external platform provides many benefits such as faster data-driven decision making, increased productivity and enhanced user experience (Mostert, 2023).

Faster Data-Driven Decision Making

The ability to provide users with the correct data insights into their design process and with suitable context can help them to make data-driven decisions faster since the data is available for the user straight away without them having to retrieve it (Mostert, 2023).

Increased Productivity

Since the data insights is available immediately and embedded in the business process, it reduces the time it takes to search for data insights through different tools which would require constant context switching and decrease productivity on more important tasks (Mostert, 2023).

Enhanced User Experience

Embedded analytics is able to provide a seamless user experience through integrating analytics into existing applications. Through easy to use and self-service data exploration capabilities, users are easily able to convert complex data into interactive visualisations and dashboards. These features help to promote engagement, data literacy and also enables users to obtain valuable insights on their own applications (Databrain, 2023).

What Do Developers Use to Implement Embedded Analytics Into Software?

The author researched an article by Tim King for Solutions Review which listed the top 20 Best Embedded Analytic Tools and BI software for 2025 (King, 2025). The author researched two of the embedded analytic services which were Bold BI and Tableau Embedded Analytics. The author researched Tableau and Bold BI websites to discover what services they can provide to users.

Bold BI (Syncfusion)

Bold BI services allow developers to embed analytics straight into their applications through their developer-friendly APIs (Bold BI, no date).

- **Seamless Embedding and Deployment:** Bold Bi allows developers to quickly and seamlessly integrate interactive dashboards, designers, and widgets into their applications through their flexible APIs and SDKs (Bold BI, no date).
- **AI-Powered Insights:** Developers have the ability to create dashboards or explore data through conversational analytics that is powered by AI-driven automation, natural language and predictive analysis to help with real-time decision-making (Bold BI, no date).
- **Collaboration Export and Responsive Experience:** Developers can collaborate with comments and can also export their dashboards to different formats such as PDF and Excel (Bold BI, no date).

Bold BI embedded analytics services are also able to be integrated on a wide variety of frameworks such as React, Angular and .NET (Bold BI, no date).

Tableau Embedded Analytics

Tableau embedded Analytics is able to provide their interactive and analytical capabilities to your products and applications seamlessly, without resource managing and also without building it from scratch (Tableau, no date). Some of the features Tableau Embedded Analytics have to offer includes:

- **Seamless Integration:** Tableau provide APIs and developer tools to easily integrate Tableau analytics into your own products, applications and web portals in a short amount of time (Tableau, no date).
- **Customizable Self-Service Analytics:** Tableau's features can be embedded into your application to supply AI-powered analytics to your users' requirements. These features include Tableau Pulse that can enable users to track KPIs through interactive dashboards, ask questions with Einstein and design visualizations to discover insights through web authoring (Tableau, no date).
- **Secure Platform:** Tableau has a REST API that enables users to automate user, content and permissions management. Tableau's Postman collection allows users

to create event-driven scripts that can allocate users, migrate workbooks, refresh data and many more (Tableau, no date).

Tableau also offer their Embedding Playground service which is an interactive learning environment to allow developers to get hands-on experience with Tableau Embedded Analytics features on their application (Tableau, no date).

2.4.3 Predictive Analytics

Predictive Analytics is the process of predicting future outcomes or trends through the use of historical data to help influence strategic decisions. Predictive analytics plays an important role in many industries since it helps to predict future events (Cote, 2021). Machine learning and deep learning techniques are utilised by data scientists to identify patterns and predict future events. (IBM, 2022).

Purposes of AI-Predictive Analytics in Software Development

According to an article by Everite Solutions, the use of AI-powered predictive analytics can be used for software development projects by analysing previous projects to help teams predict timelines, budgets, and potential risks. This helps organizations to improve their project management processes and gain more success. They listed some reasons why incorporating AI-powered predictive analytics into software development projects is important, some of the reasons included enhanced project planning, reducing risks and optimizing resource allocation (Everite, 2024).

Enhanced Project Planning

The ability to retrieve insights into potential challenges that could occur, it enables them to better plan out more realistic project timelines and also allocate resources more efficiently. Better insights allow project managers to reduce their reliance on guessing and make data-driven decisions (Everite, 2024).

Reducing Risks

The use of AI-powered predictive models enables teams to identify risks before they escalate. This allows teams to place proactive measures that can save time and resources. Identifying risks helps enhance significant cost savings and avoid setbacks that could cause projects to fail (Everite, 2024).

Optimizing Resource Allocation

Through observing predictive insights, it helps to influence teams on what resources they need for the project. This also help teams to better plan their personnel allocation and their budgets which can minimize wastage (Everite, 2024).

What Do Developers Use to Implement Predictive Analytics into Software?

The author researched an article by Muhammad Hassan which talked about the various techniques and tools that are utilised. The list included Python and Altair in which the author carried out more research into these tools (Hassan, 2024).

Python

Python is one of the most popular programming languages that is used in the software industry. It is considered to be one of the easiest languages to learn and is widely used in technological fields such as Data Science, Machine Learning and Automation Tasks (GeeksForGeeks, 2024). Python includes various libraries that are used in Data Science which includes NumPy, Pandas and Scikit-Learn.

- **NumPy:** NumPy is a widely used open-source python library for scientific computation. It consists of built in mathematical functions that is able to support multidimensional data and large matrices (Ali, 2024).
- **Pandas:** The Pandas library is commonly used for data analysis, data manipulation and data cleaning. It allows users to seamlessly perform data modelling and data analysis with very little code (Ali, 2024).
- **Scikit-Learn:** Scikit-Learn is a popular machine library in Python that is built on top of NumPy, SciPy, and Matplotlib. It is an easy and efficient tool that is used predictive analytics (Ali, 2024).

Altair

Altair offers their data analytics and AI platform called RapidMiner which provides users the ability to discover data insights, utilize data analytics and advanced AI automation for scalable solutions (Altair, no date). The author listed some features Altair have to offer.

- **Break Down Data Silos:** Altair RapidMiner allows users to merger siloed data across the user's organisations. They also offer advanced data fabric tools enables

users to seamlessly access, move and transform structured and unstructured data(Altair, no date) .

- **Access Dark Data:** Users can access inaccessible data formats such as PDFs or business reports through RapidMiner's automated data extraction tools (Altair, no date).
- **Enhanced Decision Making:** Altair provide easy to use tools for data analytics such as creating dashboards or predictive models which helps users to discover new insights and improve their decision-making (Altair, no date).

Altair RapidMiner has multiple technology partners which include Nvidia, AWS and Microsoft (Altair, no date).

2.4.4 How Do Software Developers Incorporate Data Analytics Into Software Chapter Conclusion

There is a wide variety of ways software developers incorporate data analytics into software such as data logging, embedded analytics and predictive analytics through a wide range of tools and platforms. The author found the research on embedded analytics to be quite interesting as it is similar to what the author is trying to achieve in project by implementing suitable data visualization techniques based on the sports club data to help identify trends that could be useful to the sport club.

2.5 Data Visualization

Data visualization is used to present data with complex relationships and also display data-driven insights through the use of visual graphics such as charts, plots or infographics to display the data in an easier to understand format (IBM, 2021). People utilize data visualization in order to get an understanding of their data by easily displaying patterns, trends and outliers, it also makes it easier to present your findings to non-data experts and allow people to tell a story with their data (Stevens, 2023). The author was reading an article by IBM about Data Visualization in which the author decided to research into pie charts, line charts, scatter plot data visualization techniques (IBM, 2021). The author also researched into how data visualization is used in the sport industry.

2.5.1 Pie Charts

Pie Charts are one of the most commonly used data visualization techniques. A pie chart is able to display a single variable that is split into various segments based on certain categories in the data. Each section in the pie chart should be equal to 100% when it is all

added together. The best use case for implementing a pie chart is for data with a small number of categories so each section is visible (Lin, 2023).

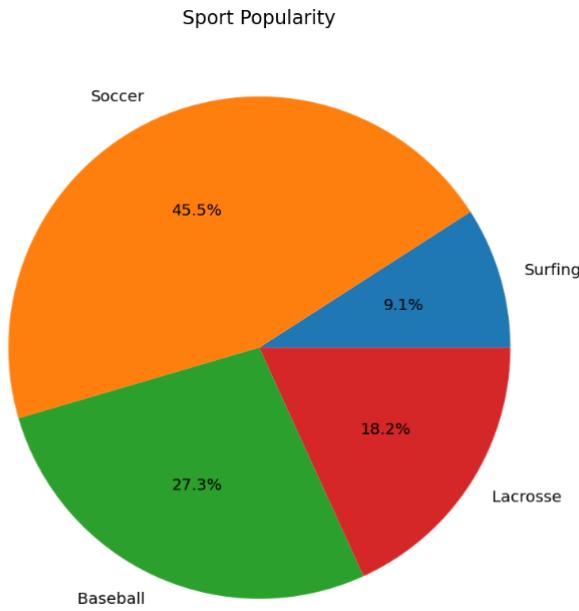


Figure 1: An Example of A Python Pie Chart To Display The Popularity of Each Sport (Alex, 2021)

Pie charts are not really a popular option as they can be hard to interpret information, being able to match the labels and the slices can be difficult and small percentages can also be difficult to display which could be vital information (The University of Melbourne, no date). A bar chart is an alternative to using pie charts, the bars can be displayed vertically or horizontally and each bar represents a slice in a pie chart. Bar charts can also display zero or negative values which isn't displayed on pie charts but it can be difficult to distinguish the difference in the tops of the bars for large values (Scutaru, no date). The author will not be using pie charts in his application and will instead use bar charts.

2.5.2 Line Charts

Line Charts are used to display the behaviour of data over a certain amount of time. The x and y axis are both numerical values in which the x-axis is the quantitative measurement and the y-axis represents the time-span. The chart consists of plotted dots which represent the data points for each time interval, these data points are connected to each other with a line which helps to identify positive or negative trends over a certain timespan. Line charts can consist of single line or multiple lines that represent multiple categories (Lin, 2023).

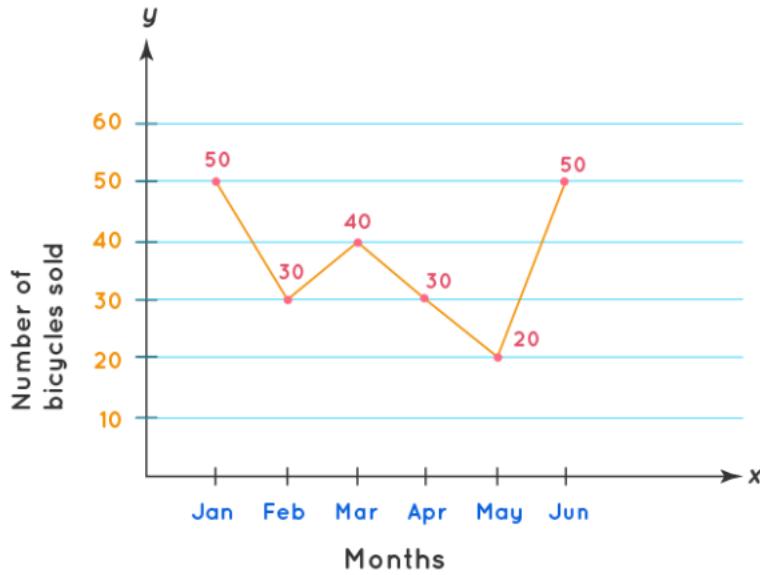


Figure 2: A Simple Line Graph Displaying The Number of Bicycles in the First 6 Months of the Year (CueMath, no date).

The author implemented line graphs using D3 that shows the number of club members that joined in each month over the past six months and also to display trends over time in a selected players height, weight, body fat percentage and body mass index.

2.5.3 Scatter Plot Graphs

Scatter plot graphs are used to display the relationship between two numerical variables, the x axis consists of one numerical variable while the y axis consists of the other numerical variable. A circle is plotted on the graph where the two numerical variables intersect. This is useful for bigger datasets that do not contain a time variable (Lin, 2023).

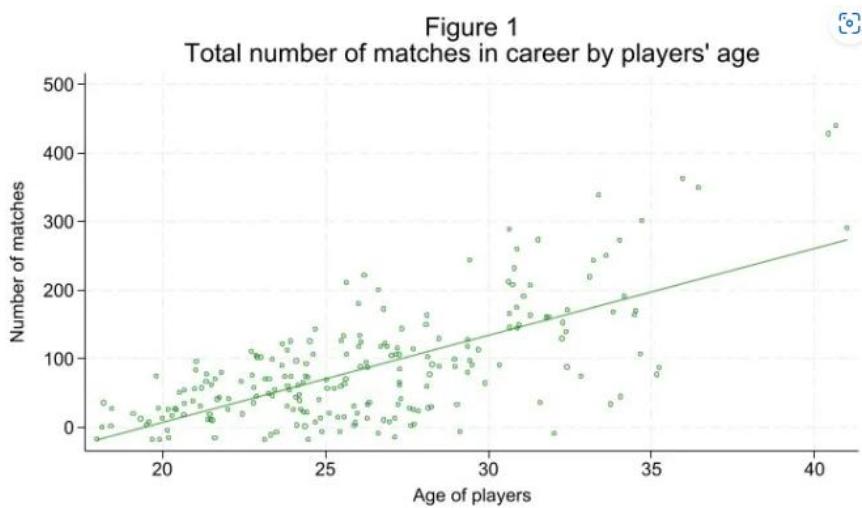


Figure 3: An example of a scatter plot being used to display the correlation between the number of matches played and the age of players (Thrane, 2023)

The author was going to implement scatter plots into his web application to display relationships between players physical attributes, but the author believed it could be hard for trainers to interpret any findings from the scatter plots as some variables might not have any correlation with each other. The author decided it would be better to implement a line chart in which it can display trends over time in the players height, weight, body fat percentage and body mass index.

2.5.4 Application Of Data Visualization in Sports

The ability to display complex statistical data can provide many benefits to sport organizations whether it's on or off the field of play. The application of data visualization tools and software to gather sports statistics can make complex data more accessible to users (Jovanovic, 2024). The author researched an article about how the Miami Heat utilised Microsoft's data visualization tool called Power BI to learn more about their own fans (Briggs, 2019).

Edson Crevecour and their team were able to forecast the game attendance for each of their 41 home games. They also forecasted other metrics such as ticket purchases, food and drink purchases, retail purchases and staff requirements necessary for each event in their arena. Through the help of Microsoft Power BI data visualization tool, the team were able to present patterns of ticket purchases, food and drink purchases and retail purchases from previous seasons which helped to generate their forecasts. Miami Heat employees were able to read the data with Power BI in order to get a better understanding of the behaviour of certain individual fans (Briggs, 2019).

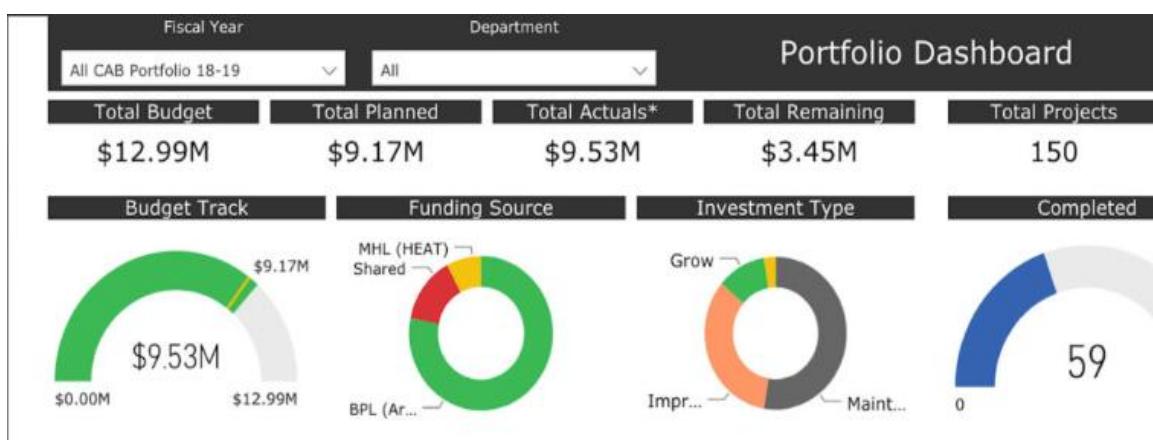


Figure 4: Power BI dashboard created by Miami Heat (Briggs, 2019).

2.5.5 Data Visualization Chapter Conclusion

The use of data visualization is very beneficial as it can display complex data in an easy-to-read format which helps with identifying patterns, outliers, certain trends, ability to tell a story with the data and also displaying your findings to non-data experts. Data Visualization techniques such as pie charts is useful for displaying single variables that contain categories but it is not a popular choice, line graphs to display the behaviour of data over a certain time period and scatter plot graphs to display the correlation between two numerical variables. Data Visualization is used by sport organizations to display complex data that can help them on and off the field of play. An example of this was the Miami Heat utilizing Microsoft's Power BI to forecast sales and find out more about their own fans. The author found it intriguing on learning more about data visualization and how it could be applied in the sports industry. Through this research, the author decided to implement bar graphs, line graphs and tables in his application to visualize data insights and trends in the sport club. Tables were used to display information such as club events, team sessions, etc .Examples of how bar charts were used in the application was to display the number of males and females members in the sports club and the top performers for each metric recorded in a team session. The author used line charts to display the number of new registered members in each month for the past 6 months and displayed player physical metrics trends over time.

2.6 Web Application Frameworks

Web application frameworks are utilized to develop web applications, web services, web resources and also web APIs in a more efficient way(GeeksforGeeks, 2024a). They consist of tools and libraries that can assist developers in developing and maintaining web applications (Kumar, 2024). Each web development framework contains different features such as pre-built modules, libraries, APIs and tools (Kumar, 2024). The author decided to research more about the Spring Boot, Django and Angular Frameworks. The author learned about using the Spring Boot framework during his final year, the author decided to research the Django framework since it is a Python framework which consists of the NumPy, Pandas, Matplotlib and Seaborn libraries that the author learned about by taking the Big Data Mining and Analysis module that was taught by James Fennell and also the Angular framework because the author has worked with this framework during his work placement.

2.6.1 Spring Boot

The Spring Boot Framework is an open-source Java MVC web application framework that is used for creating Spring-Based enterprise applications. The Spring framework was created in the early 2000s and would quickly become one of the most widely used Java frameworks for enterprise applications. The Spring Boot framework was then created which was easier to use compared to the original Spring framework. Spring Boot consists of all of the same tools and technologies that Spring has to offer which allows for rapid application development. It adds functionality to Java Applications through the use of dependency injection and inversion of control (Zeichick, 2024).

Spring Boot Features

- **Faster Development:** Spring Boot is able to find, load and set up all of the essential modules, functions and third-party libraries through using “opinionated” algorithms (Zeichick, 2024).
- **Easy To Configure:** Spring Boot can automatically configure modules and third-party libraries on your project. It is also able to update and configure modules and libraries if they are changed by the developer throughout the development process (Zeichick, 2024).
- **Dependency Management:** Spring Boot supports dependency injection that allows for external objects and functions to be integrated in the application easily, developers can focus more on the business logic of the application (Zeichick, 2024).

2.6.2 Django

Django is an open-source Python web application framework that allows developers to rapidly develop secure and maintainable web applications (MDN Web Docs, 2025). Django follows the Model View Template (MVT) design pattern.

- **Model:** The purpose of the model is to get data from the database, it delivers data in the form of an Object Relational Mapping (ORM) which makes it easier for developers to communicate with databases as long as they have a good understanding of the structure of the database. The use of object relational mappers prevents the need for developers to write complex SQL queries. (W3schools, no date)

- **View:** The purpose of the view is to handle http requests, retrieve the relevant model and figures out what data needs to be sent the template file and returns the result (W3schools, no date).
- **Template:** The template file is where developers define how the result should be displayed to users (W3schools, no date).
- **URLs:** Django retrieves URL the user requested and decides what view should be displayed according to what the developer defined in the urls.py file. (W3schools, no date)

Django Features:

- **Versatile:** It is possible to build any type of web application as it is able to cooperate with any client-side framework and also send data in any format. It also has the ability to be extended to utilise other components (MDN Web Docs, 2025)
- **Security:** Django is able to provide a secure way to manage user accounts and passwords, prevents session information from being stored in cookies, storing password hash instead of directly storing passwords and can also provide protection against SQL injection, cross-side request forgery, cross-side scripting and clickjacking (MDN Web Docs, 2025).
- **Scalability:** Django uses a shared-nothing architecture meaning that each component is independent of each other and can be easily replaced or updated. This allows for applications to be easily scaled to meet the needs of incoming traffic by adding hardware at any level (MDN Web Docs, 2025).

2.6.3 Angular

Angular is an open-source web-application framework that was created by Google. It is a component-based architecture that enables developers to build dynamic and scalable single-page applications (GeeksforGeeks, 2025).

Angular Features

- **Custom Components:** Developers can create components that contains functionality as well as rendering logic into reusable components (GeeksforGeeks, 2024b).

- **Dependency Injection:** Developers can create their own modular services and inject them to anywhere it is needed which also improves testing and reusability of the same services (GeeksforGeeks, 2024b).
- **Compatibility:** Angular is able to work on various operating systems including Windows, Linux and Mac OS. It is also able to run on all browsers (GeeksforGeeks, 2024b).

Web Application Frameworks Chapter Conclusion

The use of web application frameworks helps make it easier for developers to create web applications and other web services through the use of tools, libraries, pre-built modules and APIs that the framework has to offer. The Spring Boot framework is an open-source Java framework that supports rapid application development, automatically configures modules and third-party libraries on your project and also uses dependency injection to allow developers to focus on the business logic of the application. Django is an open-source Python web application framework which is able to provide flexibility, security and scalability. The Angular framework which is generally used to create single page applications and can allow developers to create custom components, create their own modules and inject them into their code. Angular is also compatible on numerous operating systems and browsers. The author found it helpful researching more about these three frameworks and what they have to offer. The author used the Spring Boot web application framework as the author learned more about it by taking the Enterprise Application Development, API Design & Development and Reactive Systems modules during his final year in which the author became very familiar with developing web applications and REST APIs using Spring Boot.

3 Design & Analysis

In this chapter, the author will discuss about the functional and non-functional requirements of the application, utilising Agile, use cases for each user of the application, technologies utilised in the development of the application, the components of the application, the system's architecture and the database design.

3.1 Functional Requirements

The author has experience using the ClubZap application for keeping up to date with news and upcoming events in his local GAA club. The author also downloaded the Teamer application that helps users to manage their teams and researched into the Stat Sports Apex GPS device. After using and discovering the features from the ClubZap and Teamer applications and researching the Apex GPS device, the author decided on these functional requirements for the sports club management web application.

- **Club Administrators:** club administrators should be able to login / logout, view club member statistics, create and manage club events, view club events, view more information on a selected club event, log attendance for a club event, create and manage teams, view teams, view more information on a selected team and register a new club member(club member, player, trainer or admin).
- **Club trainers:** club trainers should be able to log in/log out, view their teams, view more information on a selected team, record and view player physical statistics, view club events, view more information on a selected club event, log attendance for a club event, create and manage team sessions, view team sessions, view more information on a selected team session, view team session statistics, export statistics recorded in a team session to a PDF file, add team session statistics and upload team session statistics on a CSV file.
- **Club Players:** club players should be able to log in/log out, view club events, view more information on a club event, log attendance for a club event, view team sessions, view more information on a team session, log attendance for a team session and view their teams
- **Club Members:** club members should be able to log in/log out, view club events, view more information on a club event and log attendance for club events.

3.2 Non-Functional Requirements

The non-functional requirements of a software system place more emphasis on how the system should operate rather than the functionality it should provide. The non-functional requirements are just as important as the functional requirements of the software system because even though the system can carry out all of the functionality required, the users still need to be able to use the software system at a satisfactory level. The author selected a sample of non-functional requirements the system should have. Although these are important non-functional requirements, these have been relaxed to a certain extent as this is a proof-of-concept project in which the author focused on implementing functionality to demonstrate a comprehensive sports club management web application.

- **Performance:** The system should be able to respond to user requests efficiently.
- **Usability:** The system should be easy-to-use and also be aesthetically pleasing to all users of the system
- **Security:** It's important that users are authenticated and that proper authorization is in place.
- **Reliability:** The system should be able to perform all of its functionality over a long period of time.

3.3 Agile Development

Agile Development is a project management technique that is used in the software industry. This involves breaking the project into smaller parts that can be completed in phases called sprints. The author has previous experience of applying agile methodologies during his third-year group project where each use case was given a rating based on their priority, priority 1 use cases were completed in the first iteration, priority 2 use cases were completed in iteration 2, priority 3 use cases were completed in iteration 3 and the remaining use cases were placed in backlog. Each iteration lasted two weeks. During the author's work placement, his team carried out one-week sprints which could be extended to another week depending on the amount of work that was completed. The author loosely followed this approach for the development process of the web application. The author applied priority ratings for each use case of the web application that were ranked from the highest to the lowest. The author decided that each iteration would last for two weeks. The author was able to stick to this approach for tasks that were ranked Priority 1 and

Priority 2, but the author also had other commitments such as completing assignments for other modules in which tasks that were ranked a Priority 3 took longer than two weeks.

3.4 Use Cases

The author listed the following use cases for each user of the application and their priority rating below.

3.4.1 Club Members Use Cases

- **Log in (Priority 1):** a club member should be able to log into the application.
- **Log out (Priority 1):** a club member should be able to log out of the application.
- **View Club Events (Priority 1):** A club member should be able to view upcoming club events.
- **Log Club Event Attendance (Priority 2):** A club member should be able to log if he is going or not going to an upcoming club event.
- **View Club Event Info (Priority 2):** A club member should be able to view more information on a club event.

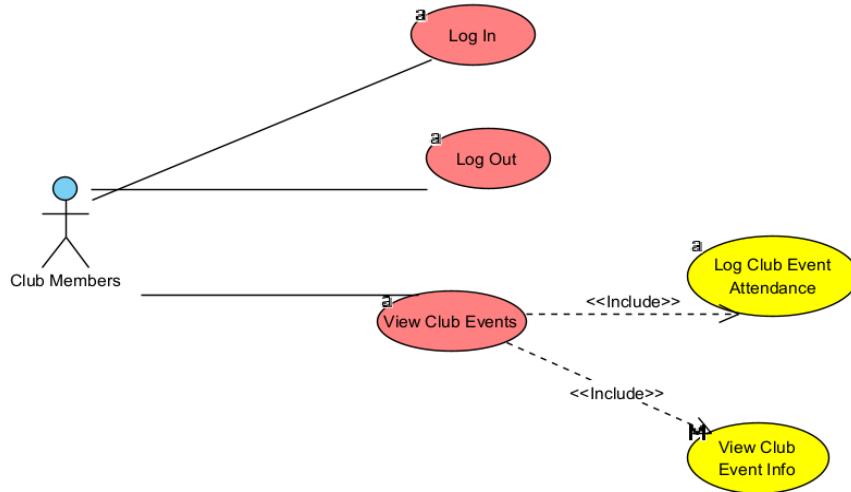


Figure 5: Club Members Use Case Diagram

Priority 1

Priority 2

3.4.2 Club Player Use Cases

- **Log In (Priority 1):** A club player should be able to log into the system.
- **Log out (Priority 1):** A club player should be able to log out of the system.
- **View Club Events (Priority 1):** A club player should be able to view upcoming club events
- **View Team Sessions (Priority 1):** A club player should be able to view upcoming team sessions that involves teams he is a part of.
- **View Teams (Priority 1):** A club player should be able to view the teams he is a part of.
- **View Team Sessions (Priority 1):** A club player should be able to view upcoming training sessions for teams they are a part of.
- **Log Club Event Attendance (Priority 2):** A club player should be able to log if they are going or not going to an upcoming club event.
- **View Club Event Info (Priority 2):** A club player should be able to view more information on an upcoming club event.
- **Log Team Session Attendance (Priority 2):** A club player should be able to log if they are going or not going to an upcoming team session.
- **View Team Session Info (Priority 2):** A club player should be able to view more information on an upcoming team session.
- **View Team Info (Priority 2):** A club player should be able to view more information on a selected team

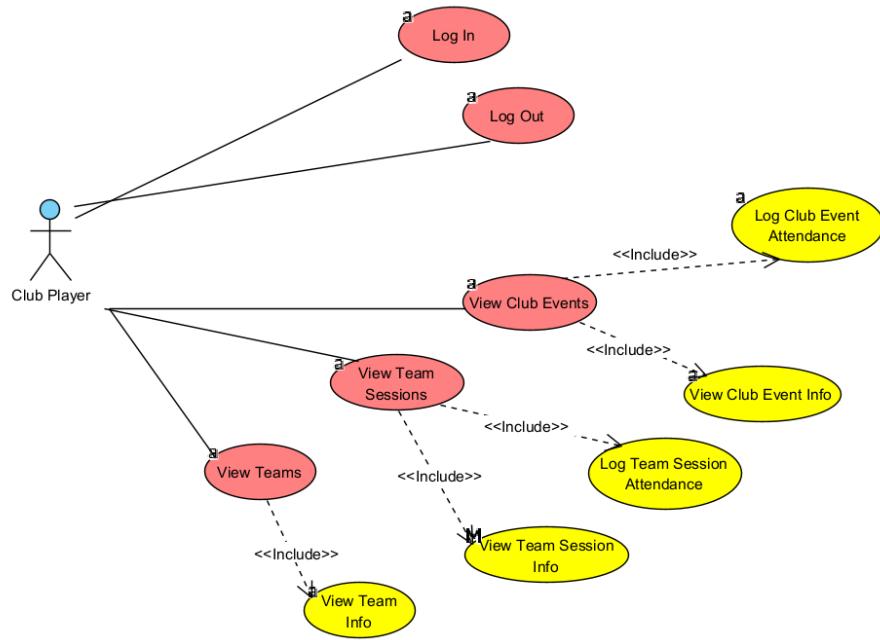


Figure 6: Club Player Use Case Diagram

Priority 1

Priority 2

3.4.3 Club Trainer Use Cases

- **Log In (Priority 1):** A club trainer should be able to log into the system.
- **Log out: (Priority 1):** A club trainer should be able to log out of the system.
- **View Teams: (Priority 1):** A club trainer should be able view the teams that they currently train.
- **View Club Events: (Priority 1):** A club trainer should be able to view upcoming club events
- **Manage Team Sessions: (Priority 1):** A club trainer should be able to create, update and remove team sessions
- **View Team Sessions: (Priority 1):** A club trainer should be able to view their upcoming team sessions.
- **View Team Info: (Priority 2):** A club trainer should be able to view more information on a selected team.
- **View Club Event Info: (Priority 2):** A club trainer should be able to view more information on a club event.

- **Log Club Event Attendance: (Priority 2):** A club trainer should be able to log if they are going or not going to an upcoming club event.
- **View Team Session Info (Priority 2):** A club trainer should be able to view more information on a team session.
- **Record Player Physical Statistics (Priority 3):** A club trainer should be able to select a player on a team and record their physical statistics.
- **View Player Physical Statistics (Priority 3):** A club trainer should be able to view the physical statistics of a selected player. This also consists of a line graph in which the club trainer can select the players height, weight, body fat percentage and body mass index to view trends over time
- **Export Team Session Statistics to a PDF file (Priority 3):** A club trainer should be able to export the stats recorded for a selected team session to a PDF file.
- **View Team Session Statistics (Priority 3):** A club trainer should be able to view the recorded statistics for a selected team session. This will also consist of a bar chart in which the club trainer can select to view the top 10 players (if there is ten or more players in the team) with the highest max BPM, average BPM, resting BPM and calories burned.
- **Add Team Session Statistics (Priority 3):** A club trainer should be able to enter the session statistics recorded from each layer that participated in the team session.
- **Upload Team Session Statistics CSV File (Priority 3):** A club trainer should be able to upload a CSV file which saves the session statistics recorded for a selected team session to the system.

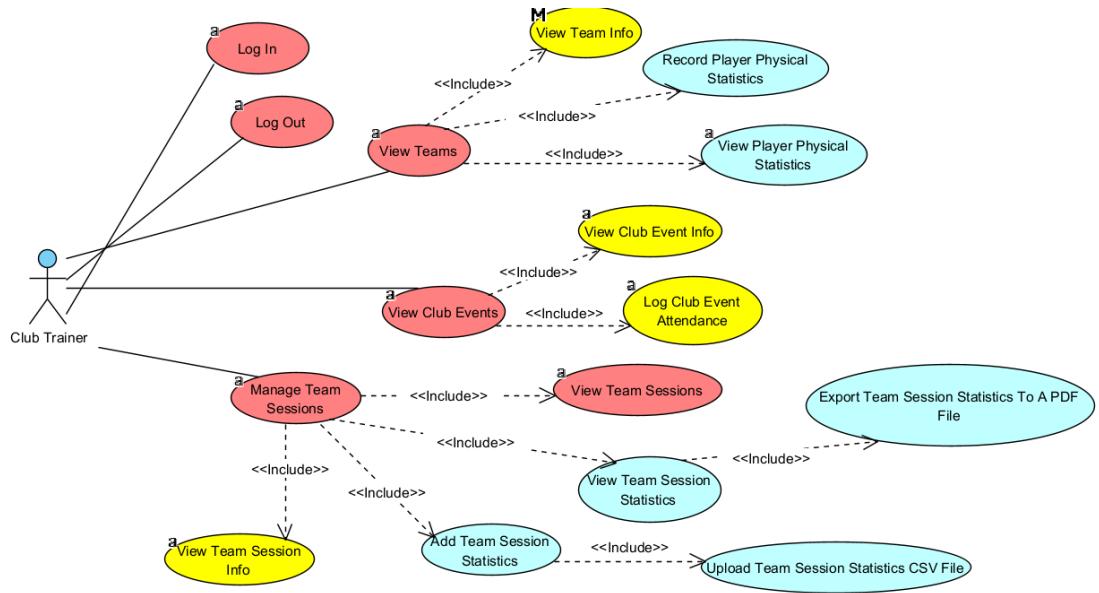


Figure 7: Club Trainer Use Case Diagram

Priority 1

Priority 2

Priority 3

3.4.4 Club Administrator Use Cases

- **Log In (Priority 1):** A club administrator should be able to log into the system.
- **Log out (Priority 1):** A club administrator should be able to log out of the system.
- **Manage Club Events (Priority 1):** A club administrator should be able to create, remove and update club events.
- **View Club Events (Priority 1):** A club administrator should be able to view upcoming club events.
- **Manage Teams (Priority 1):** A club administrator should be able to create, update and remove teams from the system
- **View Teams (Priority 1):** A club administrator should be able to view all of the teams that are recorded on the system.
- **Register A Club Member (Priority 1):** A club administrator should be able to register a new club member to the system.
- **View Club Event Info (Priority 2):** A club administrator should be able to view more information on a selected club event.

- **Log Club Event Attendance (Priority 2):** A club administrator should be able to log if they are going or not going to an upcoming club event.
- **View Team Info (Priority 2):** A club administrator should be able to view more information on a team.
- **View Club Member Statistics (Priority 3):** A club administrator should be able to view statistics on sports club members. This will consist of a bar chart displaying the number of male and female members that are registered in the sports club, the club administrator can also view how many male and female admins, players, trainers or just ordinary club members. It also consists of a grouped bar chart that displays the number of male and female members within certain age groups and also a line chart that displays the number of new members that have been registered in the sport club in each month for the last six months.

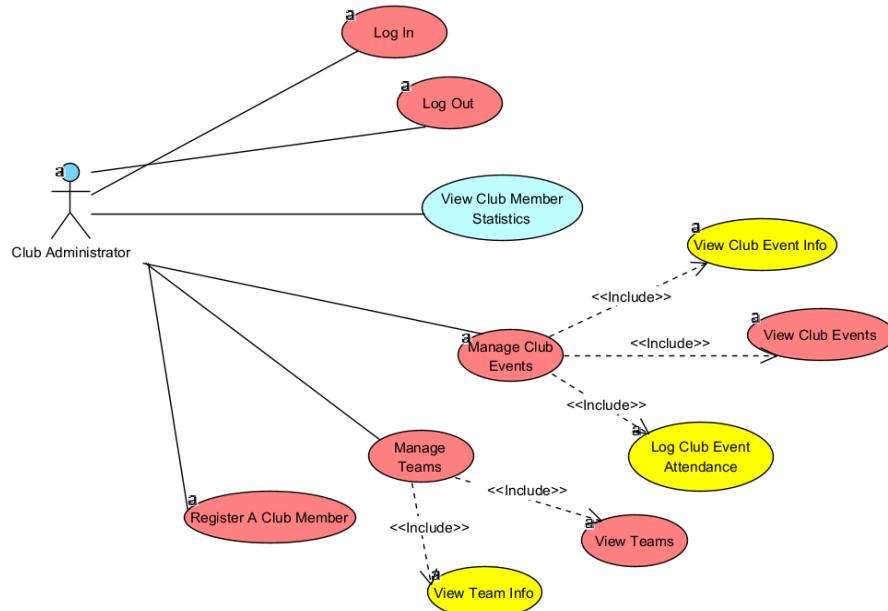


Figure 8: Club Administrator Use Case Diagram

Priority 1

Priority 2

Priority 3

3.5 Technology Stack Utilised in the Project

Spring Boot Framework

To implement the sports club management web application, the author decided to use the Spring Boot framework which is a Java web application framework. The author selected the Spring Boot web application framework as he had three modules in his final year in which they learned how to develop web applications and REST APIs using the Spring Boot Framework. The author found it very easy to set up a Spring Boot project and be able to develop most features of the web application in a short amount of time which provide him with more time to focus on more complex features of the application.

Thymeleaf

The author utilised Thymeleaf for the front end of the application which is a Java server-side template engine. The author learned about using Thymeleaf with Spring Boot in two modules during his final year. The author decided to use the Thymeleaf framework as he found it easy to set up with Spring Boot and develop HTML pages to display content to the user.

MySQL

The author used MySQL as their database to store and manage data for the application. The author has previous experience of using MySQL as a database for web applications he has developed throughout his four years at TUS. The author was able to use XAMPP to host his MySQL database on his local machine.

D3

D3 stands is an open-source JavaScript library which is used for designing graphics to visualise data. It was created by Mike Bostock in 2011 and has been used to develop award winning visualizations (D3, no date). The author learned about using D3 by taking the Data Visualization module that was taught by James Fennell and by reviewing sample graphs developed in D3 helped him to implement interactive graphs into his web application.

Bootstrap

Bootstrap is a front-end framework that is used to develop mobile web sites. The author utilised Bootstrap 3 to easily style a neat looking navigation bar and also design most of the tables in the application.

AJAX

AJAX is a web development technique that enables web applications to retrieve content from the server through asynchronous HTTP requests, the content retrieved from the server can be used to update certain parts of the page without updating the full page (MDN Web Docs, no date). This came in useful for the author as he utilised AJAX to edit the members on the team without refreshing the whole page which would cause the updated values entered by the club trainer to return to it's previous state.

Jackson

Jackson is a Java data processing library which is used to convert Java objects to JSON and JSON to Java objects. The author utilised the Jackson library to be able to convert Java objects to JSON string format which could then be parsed into JSON. This allowed the data to be accessible for D3 to implement the graphs in the application.

Apache Commons CSV

The author used the Apache Commons CSV library by Apache Commons. It is a Java library that allows developers to read and write files in CSV format. The author utilised this library to allow a club trainer to upload a CSV file containing the statistics recorded for a team session and be saved to the system.

iText Core

iText Core is an open-source Java and .NET library by iText which enables developers to create and modify PDF files. The author used this library to allow trainers to export statistics recorded from a team session to a PDF file through a REST API.

3.6 Sports Club Management Web Application Components

During the development process, the author hosted the application on his own machine. Spring Boot has an embedded Tomcat server that hosts the Spring Boot application. The author used XAMPP to launch a MySQL server on his own machine to host his MySQL database.

The web application also consists of a REST API that will produce a PDF file of the statistics recorded from a team session selected by the user.

Unfortunately, the author wasn't able to retrieve an Apex wearable GPS device from Stat Sports but decided to implement a feature that allowed users to upload team session statistics through a CSV file as a proof of concept. A club trainer could collect the data recorded from a team session, the trainer could then sync the data to the Sonra Lite web application in which they can export the data to a CSV file. The CSV file can then be uploaded to the sports club management application and record the team session statistics to the sports club MySQL database. Even though the author didn't get the Apex GPS device, the author thought it would be suitable to include the device in the component diagram and explain how trainers could potentially record data from players that participated in a team session and export the data to a CSV file.

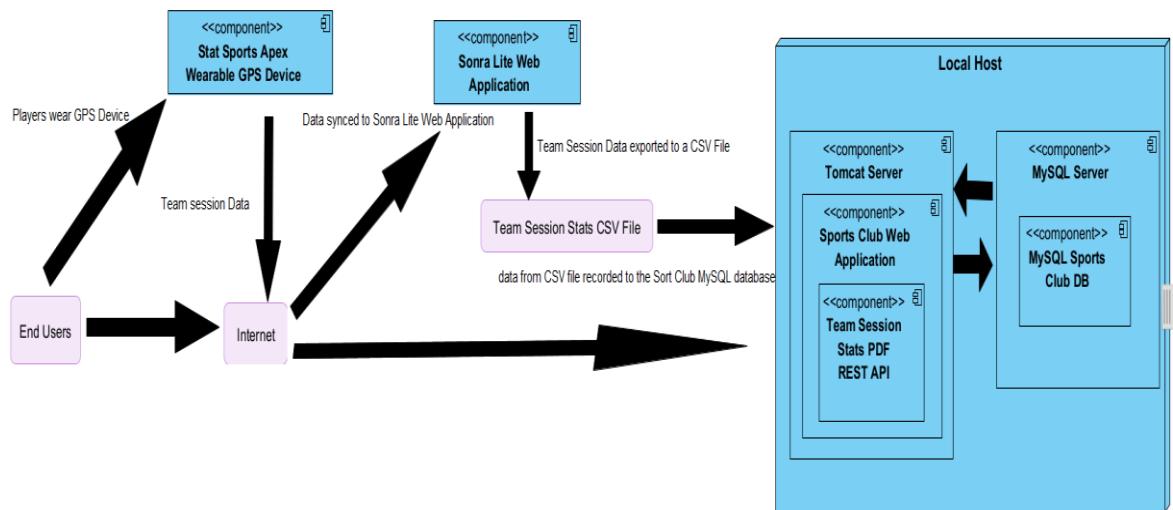


Figure 9: Sports Club Management Application Component Diagram

3.7 System Architecture

The author used the Spring Boot web application framework with the Thymeleaf template engine to design the web pages on the application. It will consist of a repository layer, service layer, controller layer, REST controller layer and Templates.

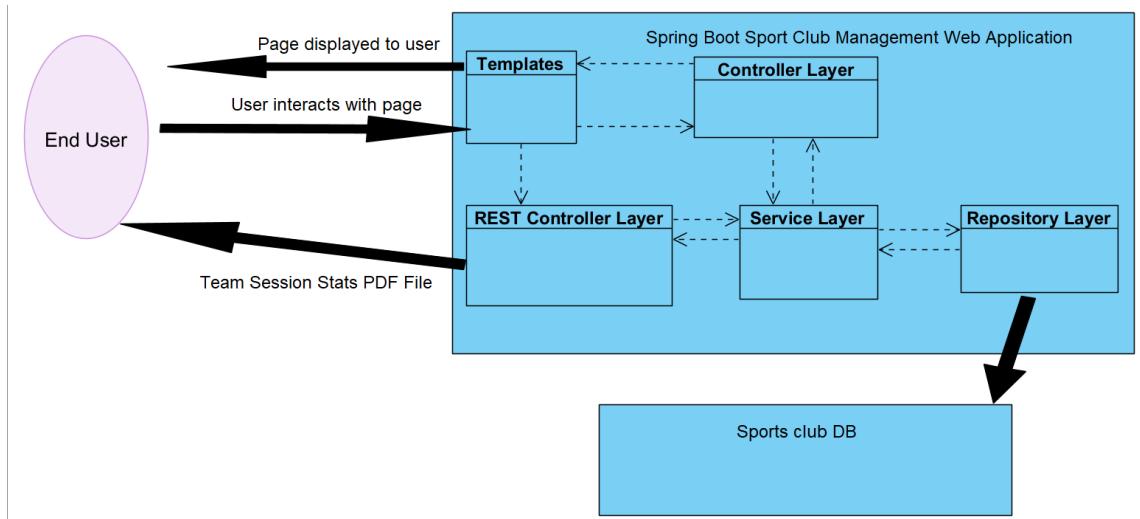


Figure 10: Sports Club System Architecture Diagram

Repository Layer: This layer interacts with the database to retrieve, update and insert objects in the sports club database.

Service Layer: This layer is where all the business logic for the sport's club management application is implemented. It retrieves the entities from the repository layer and processes them before returning them to the controller layer. It also retrieves data from the controller layer and processes them before sending it to the repository layer to save, update or remove entities. The service layer interacts with the REST controller layer by creating the contents of the PDF file, the service layer returns the PDF file to the REST controller layer so that it can send a response back to the user with the PDF file that contains the recorded team session statistics

Controller Layer: This layer handles the requests and determines which HTML view should be displayed to the user.

REST Controller Layer: This layer consists of the REST API which produces a PDF file of the statistics that were recorded from a selected team session. The trainer interacts with the HTML page to make a GET request to the REST API endpoint which returns a PDF file of the selected team session statistics as a response.

Templates: This consists of all the HTML pages in the application. These are presented to the user to display content and also allow the user to interact with the web application.

3.8 Database Design

The database was designed to accommodate the functional requirements of the web application. The author utilised MySQL as the applications database as he had previous experience of using it with the Spring Boot web application framework.

The sport club database contains 11 tables which consists of Users, User roles, Roles, Player Physical Stats, Team Session Stats, Team Events, Teams, Team Members, Team Event Attendance, Club Events and Club Event Attendance.

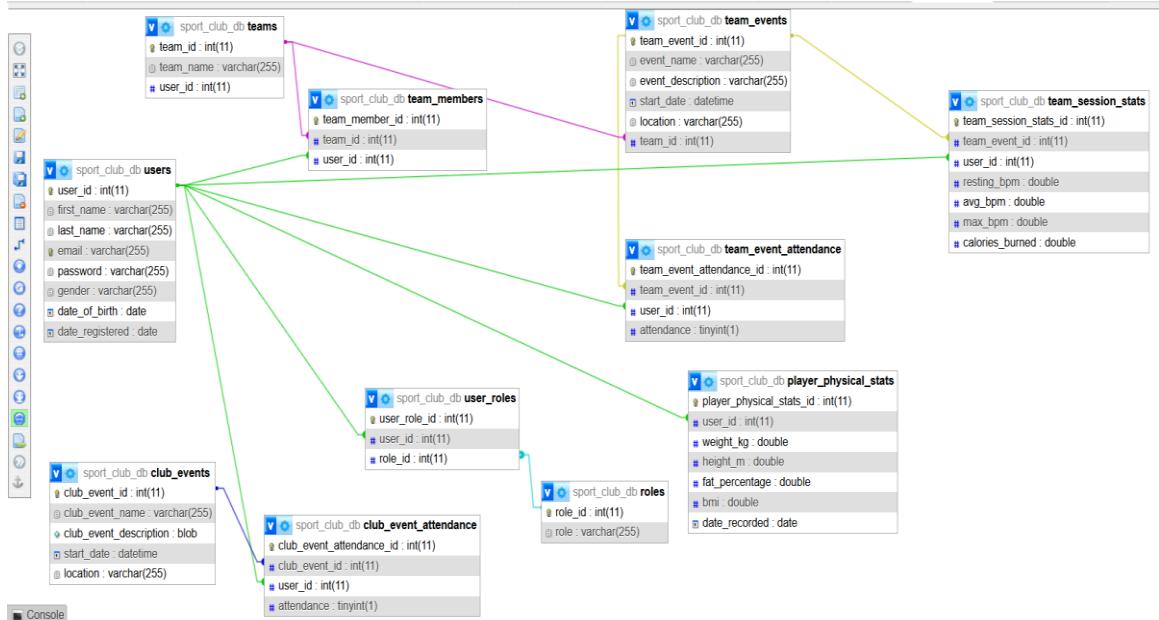


Figure 11: Sport Club DB database design

- **Users:** This table stores the information about the users of the sport club management system.
- **User Roles:** This table stores the information about the role each user has (Member, Player, Trainer, Admin)
- **Roles:** This table stores the roles the system has (Member, Player, Trainer, Admin)
- **Player Physical Stats:** This table stores the physical statistics recorded for a player (users that have the player role)

- **Team Session Stats:** This table stores the users' statistics from a team session (if the users role is a player).
- **Team Events:** This table stores information about team sessions (training sessions, matches etc.).
- **Teams:** This table stores information about teams in the sports club.
- **Team Members:** This tables stores information about team members on each of the teams in the sports club (join table to manage users and teams many to many relationship).
- **Team Event Attendance:** This table stores information about the attendance for team sessions.
- **Club Events:** This table stores information about club events taking place in the sports club.
- **Club Event Attendance:** This table stores information about the attendance for club events.

4 Implementation

The implementation chapter consists of the author discussing how he implemented the core features of the sports club management application. The author included snapshots of the code he wrote to explain how it works.

4.1 Spring Boot Security

The Spring Boot Security framework that is included in the author's Spring Boot application provides authentication and authorization to Spring Boot applications. The author learnt about this through taking the Enterprise Application Development that was taught by Alan Ryan. For one of his assignments that was assigned to the author, Alan Ryan provided code in how to set up Spring Security in Spring Boot. The author used this code and modified it according to the requirements of the author's project. The provided code consisted of the "WebSecurityConfig" class, the "loadUserByUsername", "getAuthorities" method in the User service layer, the "PasswordHasingController" and "PasswordHashingService" which is only used once to hash all the passwords in the database using Bcrypt and the "getLoggedInEmail" static method in the user class which retrieves the email of the current logged in user.

The author created a Web Security Config class which consists of two Java beans called the "filterChain", "authenticationManager". The "filterChain" method is where security features are performed by Spring Boot. This allows the author to define which endpoints are accessible depending on the users role.

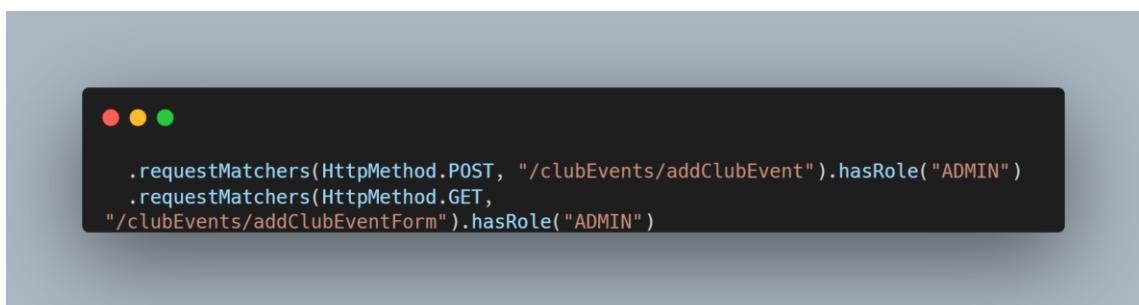
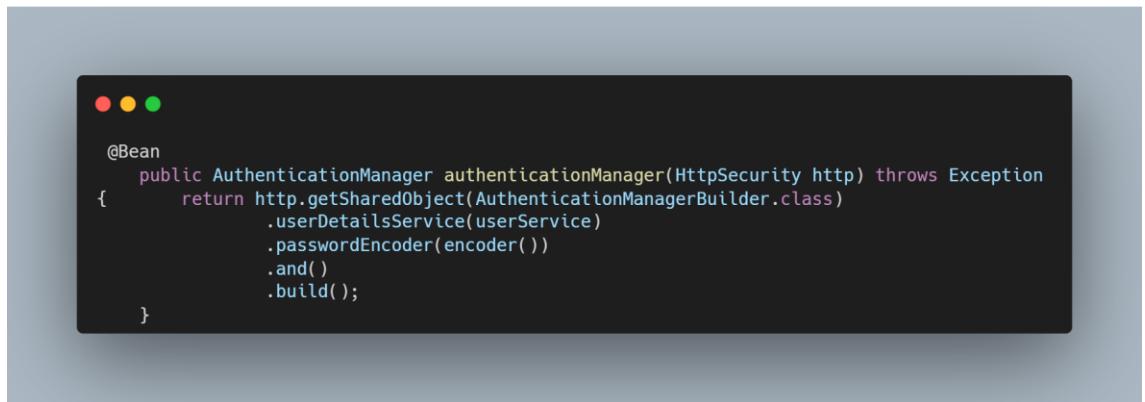


Figure 12: Example of Endpoints That Admins Can Make Requests To

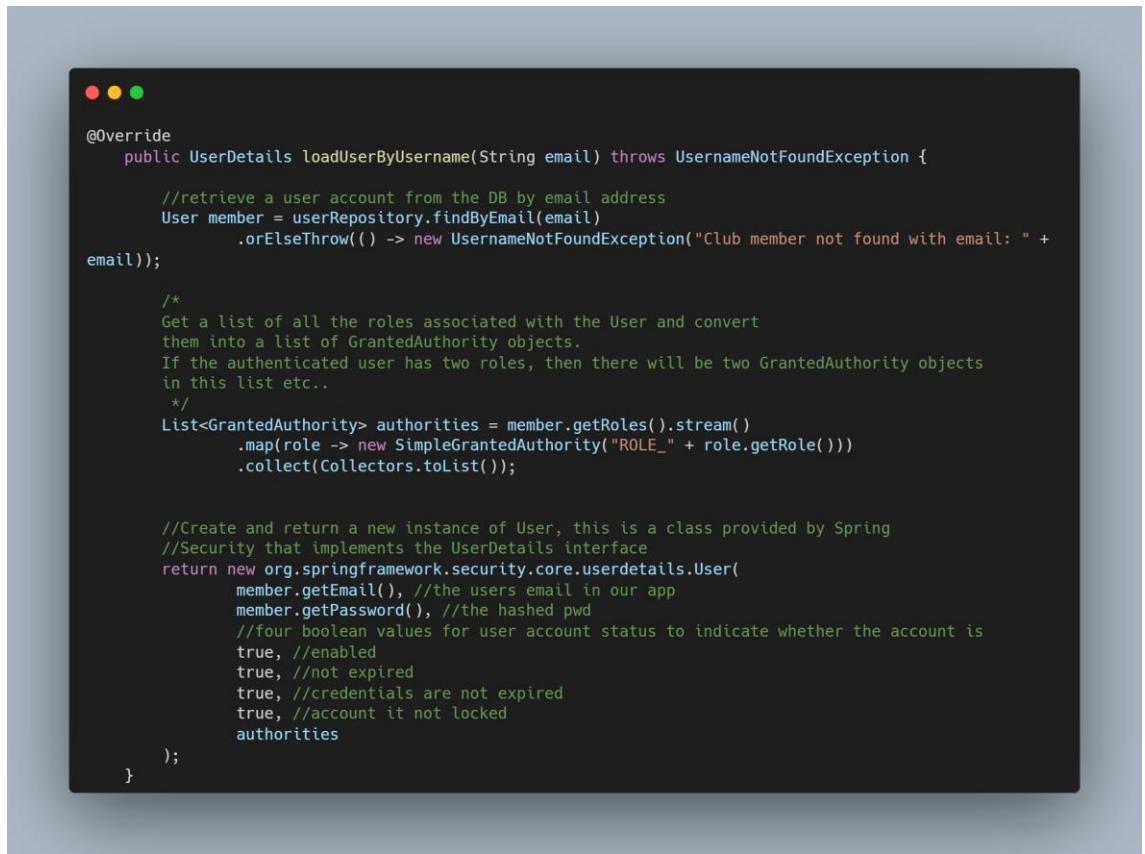
In the above example, Club administrators are the only users that can access the add club event form through a GET request and create a new club event through a POST request. These are defined within the HTTP Security object. The "authenticationManager" bean is used to verify the users' details when they attempt to login.



```
● ● ●
@Bean
    public AuthenticationManager authenticationManager(HttpSecurity http) throws Exception
    {
        return http.getSharedObject(AuthenticationManagerBuilder.class)
            .userDetailsService(userService)
            .passwordEncoder(encoder())
            .and()
            .build();
    }
```

Figure 13: Authentication Manger Bean in the Web Security Config Class

The User Service class implements the User Details Service interface in which the “loadUserByUsername” is overridden. This method tries to locate the user by the email that was entered and applies the associated roles belonging to the user.



```
● ● ●
@Override
public UserDetails loadUserByUsername(String email) throws UsernameNotFoundException {

    //retrieve a user account from the DB by email address
    User member = userRepository.findByEmail(email)
        .orElseThrow(() -> new UsernameNotFoundException("Club member not found with email: " +
email));

    /*
    Get a list of all the roles associated with the User and convert
    them into a list of GrantedAuthority objects.
    If the authenticated user has two roles, then there will be two GrantedAuthority objects
    in this list etc..
    */
    List<GrantedAuthority> authorities = member.getRoles().stream()
        .map(role -> new SimpleGrantedAuthority("ROLE_" + role.getRole()))
        .collect(Collectors.toList());

    //Create and return a new instance of User, this is a class provided by Spring
    //Security that implements the UserDetails interface
    return new org.springframework.security.core.userdetails.User(
        member.getEmail(), //the users email in our app
        member.getPassword(), //the hashed pwd
        //four boolean values for user account status to indicate whether the account is
        true, //enabled
        true, //not expired
        true, //credentials are not expired
        true, //account is not locked
        authorities
    );
}
```

Figure 12: Load User By Username Method.

With the Spring Security Framework, the author was able to utilise the JSP tag library which helped the author to hide information and certain functions of the application depending on the user's role.

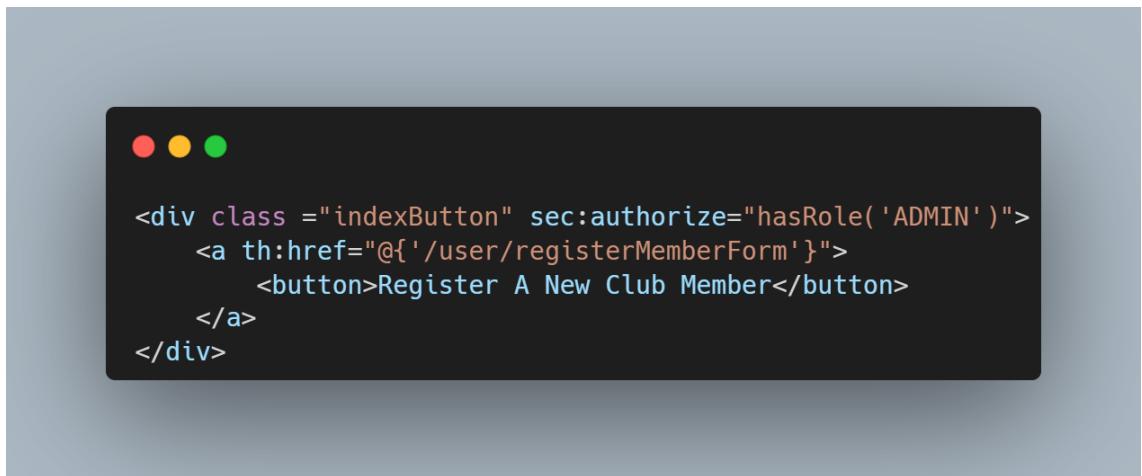


Figure 14: Application of a spring security JSP tag in HTML code

In the example above, the author was able to display certain links in the application depending on the role of the user. The “Register A New Club Member” button will only appear if the user is an Admin.

4.2 Utilising Thymeleaf Fragments

The author decided to use the Thymeleaf template engine as it works seamlessly with Spring Boot and it was able to support rapid development which was crucial for the author as he had a busy schedule.

The author noticed that he had to constantly update the navigation bar in every HTML file in the application whenever the author added a new feature. The author found this process to be very tedious and decided to research Thymeleaf features that would allow the author to implement reusable HTML code and enable him to make modifications on a single HTML file. Thankfully, the author discovered Thymeleaf fragments which allowed the author to reuse common HTML code in his application.

The author created a fragments folder within the templates directory, the fragments folder would contain all the HTML files that were going to be reused.

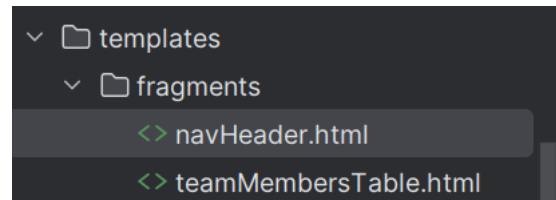


Figure 13: Fragments file structure

The author created a new HTML file called “navHeader” which included the navigation bar HTML code. In the “navHeader” HTML file, the author wrapped the navigation bar code in a div and included the “fragment” attribute to make it a fragment called “navigation-header”.



Figure 14: Fragment tag for navigation bar.

The author was then able to include this fragment into all of the other HTML template files. The author used the “insert” tag which would include the navigation bar within the div tags.



Figure 15: Insert tag to insert navigation-header fragment.

The navigation bar would then be included into the HTML page when it is rendered. The author found the application of Thymeleaf Fragments in his project to be very useful as he only needed to modify one file to add a new feature to the navigation bar instead of multiple HTML files. This also reduced the risk of forgetting to add new features in the

navigation-bar in some of the HTML files which would have made the application inconsistent.

4.3 User JPA Entity Example

Spring Boot supports Object Relational Mapping which allowed the author to work with objects instead of writing SQL queries to create, read, update or delete data from the database. Each JPA entity represents a table in the database and each JPA instance represents a row in that table. The author created a user JPA entity class which would represent all members in the sport clubs.

```
● ● ●

@Entity
@Table(name = "users")
public class User implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "user_id")
    private Integer userId;

    @Basic(optional = false)
    @Column(name = "first_name")
    private String firstName;

    @Basic(optional = false)
    @Column(name = "last_name")
    private String lastName;

    @Basic(optional = false)
    @Column(name = "email")
    private String email;

    @Basic(optional = false)
    @Column(name = "gender")
    private String gender;

    @Basic(optional = false)
    @Column(name = "password")
    private String password;

    @Basic(optional = false)
    @Column(name = "date_of_birth")
    @DateTimeFormat(pattern="yyyy-MM-dd")
    private LocalDate dateOfBirth;

    @Basic(optional = false)
    @Column(name = "date_registered")
    @DateTimeFormat(pattern="yyyy-MM-dd")
    private LocalDate dateRegistered;

    /*
     * Defined a many-to-many relationship between the User
     * entity and the Role entity, specifying that each club member could have
     * multiple roles and each role can be assigned to multiple club members.
     */
    @ManyToMany(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    @JoinTable(
        name = "user_roles",
        joinColumns = @JoinColumn(name = "user_id"), //The foreign key column in the join table
        user_roles) that refers to the User entity.
        inverseJoinColumns = @JoinColumn(name = "role_id"))
        //The foreign key column in the join table that refers to the Role entity.
        //I'm using a Set here so that roles are unique
        //This prevents duplicate entries for the same role on a user.
        private Set<Role> roles = new HashSet();

        //list of teams for a users with a 'PLAYER' role. These users are players in the sport club
        @ManyToMany(mappedBy = "teamMembers")
        private Set<Team> listOfTeams = new HashSet();
        //list of teams for club member with a 'TRAINER' role. These users are trainers in the sport club.
        @OneToOne(cascade = CascadeType.ALL, mappedBy = "trainer")
        @ToString.Exclude
        private Set<Team> trainersListOfTeams;

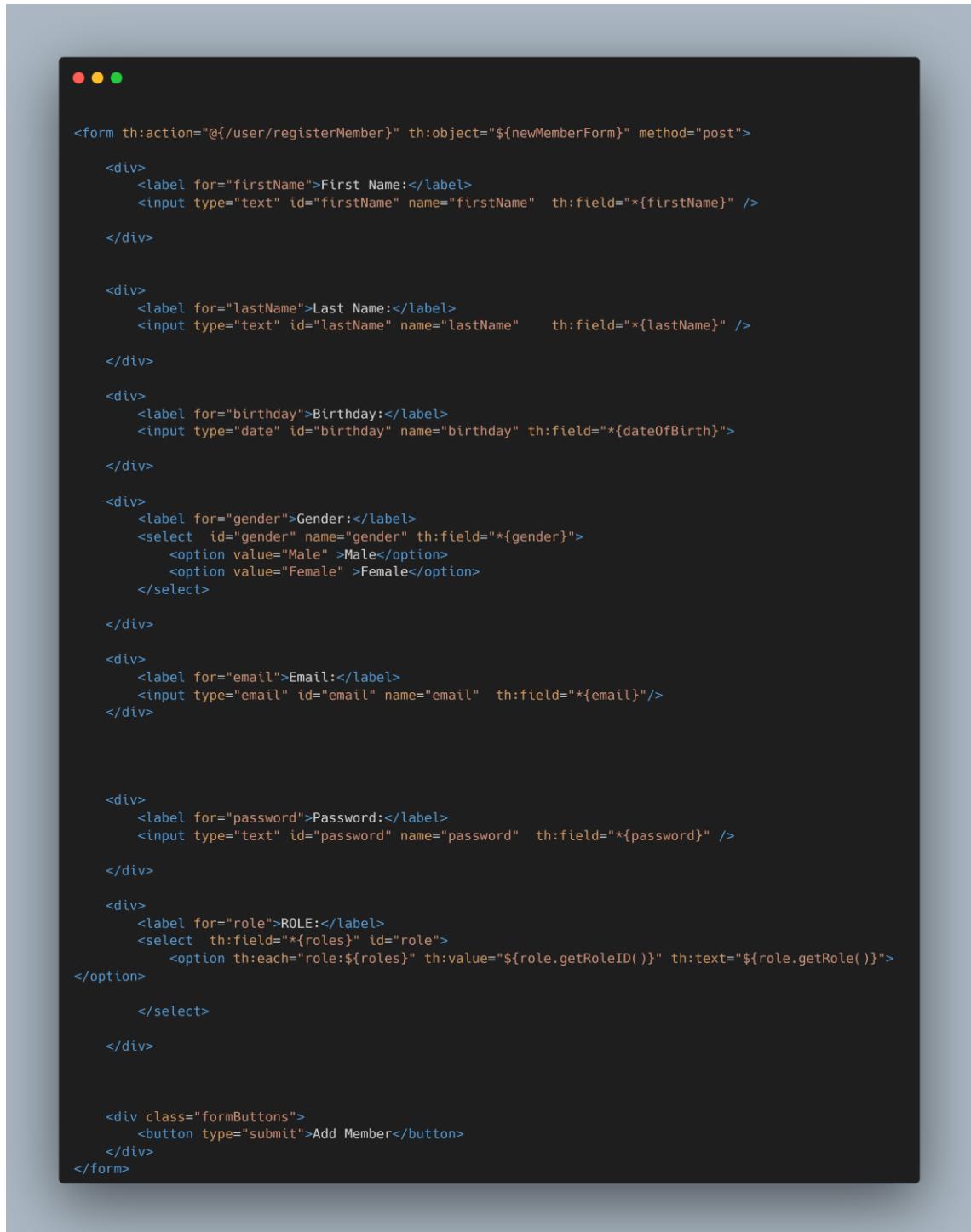
        @OneToMany(cascade = CascadeType.ALL, mappedBy = "player")
        @ToString.Exclude
        private Set<TeamSessionStats> listOfPlayerSessionsStats;

        @OneToMany(cascade = CascadeType.ALL, mappedBy = "player")
        @ToString.Exclude
        private List<PlayerPhysicalStats> listOfPlayerPhysicalStats;
```

Figure 16: User Entity JPA Class

In figure 15, the “@Entity” annotation states that the user class is a JPA entity and each instance of this entity should be stored in the database. The “@Table” annotation states that the User entity is stored in the “users” table in the database. The “@Id” annotation states that the “userId” attribute is the primary key in the database. The “@Column” annotation states which column in the “users” table that the attribute should be stored in. The user entity has a many to many relationship with the role entity, The “@JoinTable” annotation is used to define that the “user_roles” is the join table for the many to many relationship between the user and role entities. The “@JoinColumn” annotation is used to define the foreign key columns in the “user_roles” join table for the User and Roles entities. There is also a many to many relationship between the user and team entities for users that are players. The “mappedBy = teamMembers” states that the relationship is managed by the “teamMembers” attribute in the team entity. The “@OneToMany” annotation states the user entity has a one-to-many relationship with the team entity, The one-to-many relationship with teams are for users who are trainers in the sport club.

4.4 Registering A Member ThymeLeaf Form.



The screenshot shows a Mac OS X application window with a dark gray background. At the top left are three colored window control buttons (red, yellow, green). The main content area contains the Thymeleaf template code for a member registration form. The code uses the Thymeleaf syntax to bind form fields to Java objects. It includes fields for First Name, Last Name, Birthday, Gender (Male/Female), Email, Password, and Role selection. A submit button is also present at the bottom.

```
<form th:action="@{/user/registerMember}" th:object="${newMemberForm}" method="post">

    <div>
        <label for="firstName">First Name:</label>
        <input type="text" id="firstName" name="firstName" th:field="*{firstName}" />
    </div>

    <div>
        <label for="lastName">Last Name:</label>
        <input type="text" id="lastName" name="lastName" th:field="*{lastName}" />
    </div>

    <div>
        <label for="birthday">Birthday:</label>
        <input type="date" id="birthday" name="birthday" th:field="*{dateOfBirth}" />
    </div>

    <div>
        <label for="gender">Gender:</label>
        <select id="gender" name="gender" th:field="*{gender}">
            <option value="Male" >Male</option>
            <option value="Female" >Female</option>
        </select>
    </div>

    <div>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" th:field="*{email}" />
    </div>

    <div>
        <label for="password">Password:</label>
        <input type="text" id="password" name="password" th:field="*{password}" />
    </div>

    <div>
        <label for="role">ROLE:</label>
        <select th:field="*{roles}" id="role">
            <option th:each="role:${roles}" th:value="${role.getRoleID()}" th:text="${role.getRole()}">
        </option>
        </select>
    </div>

    <div class="formButtons">
        <button type="submit">Add Member</button>
    </div>
</form>
```

Figure 17: Register A Member Thymeleaf Form



```
public class RegisterMemberDTOForm {  
    private Long userId;  
    private String firstName;  
    private String lastName;  
    private String email;  
    private String password;  
    private String gender;  
    @DateTimeFormat(pattern="yyyy-MM-dd")private LocalDate dateOfBirth;  
    private Set<Long> roles;
```

Figure 18: Register Member DTO Form Class

In the “registerMemberForm” view, the “th:action=@{/user/registerMember}” defines where the author wants to send the form to which is the “/user/registerMember” controller endpoint which invokes the “addClubMember” method in the user controller layer to add a new member to the system. The “th:object” points to the model object which is “newMemberForm” that was passed from the user controller layer when it was requested to display the form. The “newMemberForm” object is of type “RegisterMemberDTOForm”. The author had to create a “RegisterMemberDTOForm” class as the user attribute “roles” is a set of “Role” objects and objects can’t be mapped to HTML form fields. The class contains all of the necessary details that the club administrator needs to provide to register a new member. The Role attribute in the “RegisterMemberDTOForm” class is of type “Set<Long>” instead of type “Set<Role>” in which the role ID can be mapped to the HTML form field since it is of type “Long”. The author also added a list of “Role” objects to the model which can be accessed in the HTML form. The “th:each” iterates through the set of roles and adds them as options within the select tag so a club administrator can select the role type of the member in the drop-down menu. The “th:field” points to the attributes of the “newMemberForm” object, the values can then be mapped to the “newMemberForm” attributes and sends it to the

“addClubMember” method in the user controller layer when the club administrator submits the form.

4.5 User Controller Layer Method Example

A screenshot of a code editor window showing Java code. The code defines a method named 'addClubMember' with the following annotations and logic:

```
@PostMapping("/registerMember")
public ModelAndView addClubMember(@ModelAttribute("newMemberForm") RegisterMemberDTOForm
clubMember,
                                    RedirectAttributes redirectAttributes) throws ParseException {
    userService.createUser(clubMember);
    redirectAttributes.addFlashAttribute("message", "Club member account has been created");
    return new ModelAndView("redirect:/");
}
```

The code uses Spring MVC annotations to handle POST requests to the '/registerMember' endpoint. It takes a 'RegisterMemberDTOForm' object from the model and calls the 'createUser' method on the 'userService'. It then adds a temporary success message to the session attributes and returns a 'ModelAndView' object that redirects the user back to the root endpoint ('/').

Figure 19: Add Club Member Controller Method

In the add club member method, the “@PostMapping” annotation states that the controller method handles POST requests. The “RegisterMemberDTOForm” instance “clubMember” is passed into the function which contains all of the necessary details entered in the form by the club administrator. The “RedirectAttributes” interface is used to redirect the user to a different endpoint in which the author can add a temporary success message to the requested end point. The message would then disappear once the page is refreshed. The controller returns a “ ModelAndView” object which redirects the user back to the “\” endpoint which is the home page of the application.

4.6 Create User Service Layer Method



```
public void createUser(RegisterMemberDTOForm member) throws ParseException {
    User newClubMember = new User();
    newClubMember.setFirstName(member.getFirstName());
    newClubMember.setLastName(member.getLastName());
    newClubMember.setDateOfBirth(member.getDateOfBirth());
    newClubMember.setPassword(member.getPassword());
    newClubMember.setEmail(member.getEmail());

    for (Long roleId : member.getRoles()) {
        Role role = roleRepository.getById(roleId);
        newClubMember.getRoles().add(role);
    }
    PasswordEncoder encoder = new BCryptPasswordEncoder();
    newClubMember.setPassword(encoder.encode(member.getPassword()));
    newClubMember.setGender(member.getGender());
    newClubMember.setDateRegistered(LocalDate.now());
    userRepository.save(newClubMember);
}
```

Figure 20: Create User Service Method

The “RegisterMemberDTOForm” member object is passed into method which contains all of the new member details entered by club administrator. A new user object called “newClubMember” is created. The author was then able to set the values of the “newClubMember” attributes by retrieving the attribute values from the member object passed into the function. The roles attribute of the member object is a set of “Long” values which is the ID of the role selected by the club administrator. The author was able to loop through the set and retrieve the role object from the “getById” method in the role repository layer. The role of the new member was then set to the “newClubMember” object. The new club members password is hashed using BCrypt. The new member is passed into the save method in the user repository layer which saves the new member to the system. The user repository is an interface that inherits the JPA Repository interface. The JPA Repository contains the save method which is used to save JPA entities to the database.



```
● ● ●  
@Repository  
public interface UserRepository extends JpaRepository<User, Integer>
```

Figure 21: User Repository Interface

4.7 Designing the Navigation Bar and Tables Using Bootstrap

The author utilised Bootstrap 3 to design the navigation bar and most of the tables that were implemented into the application.

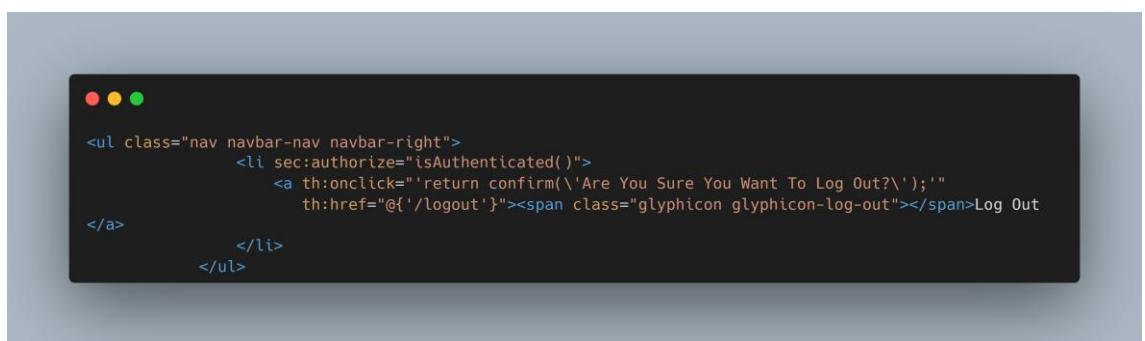
Navigation Bar Design



```
● ● ●  
<nav class= "navbar navbar-inverse navbar-fixed-top" >
```

Figure 22: Navigation Bar Bootstrap Classes

The author was able to create a black inverted navigation bar by including the “navbar-inverse” class. The “navbar-fixed-top” class enables the navigation bar to stay at the top of the screen as the user scrolls down the page.



```
● ● ●  


- <span class="glyphicon glyphicon-log-out"></span>Log Out

```

Figure 23: Log Out Button Design using Bootstrap

The “navbar-right” class allowed the author to place the log out button on the right side of the navigation bar. The author was also able to place a log out glypcan through the “glyphicon-log-out” class.



Figure 24: Navigation Bar

Table Design



Figure 25: Table Bootstrap Classes

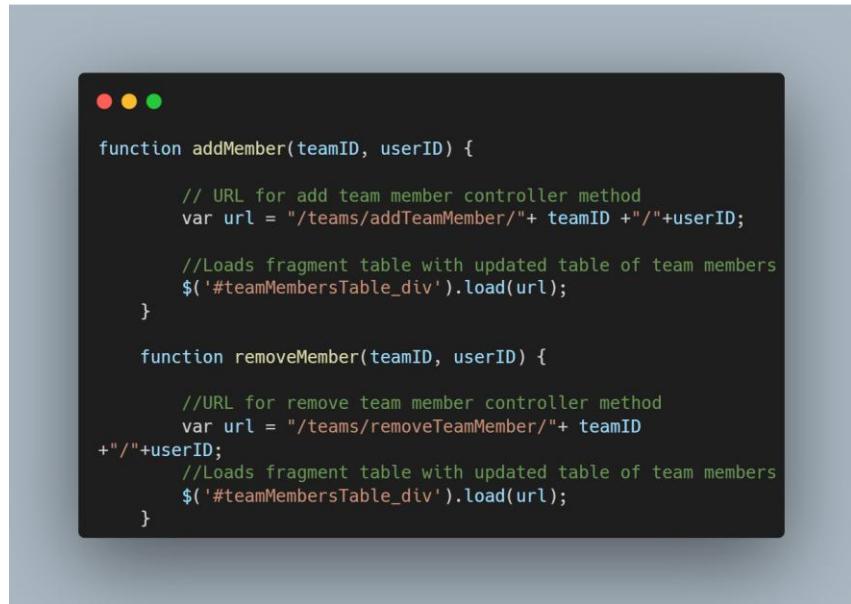
The “table-hover” class makes the background colour of a table row turn to grey when a user hovers over it.

Event Name	Event Location	Start Date And Time	Going	Not Going	Edit	Remove
dinner dance	Castle Oaks Hotel	2025-08-22 20:30	<button>Going</button>	<button>Not Going</button>	<button>Edit</button>	<button>Remove</button>
Hurling Final	Cusack Park, Ennis	2025-02-17 09:00	<button>Going</button>	<button>Not Going</button>	<button>Edit</button>	<button>Remove</button>

Figure 26: Club Events Table Example

4.8 Utilising AJAX For Editing Team Details

To edit a team, A club administrator should be able to edit the team’s name, trainer and the team members. The author wanted the web application to enable the club administrator to remove or add team members on the team without refreshing the full web page. Refreshing the full web page could revert the updated team’s name and the new trainer that the club administrator entered to its previous state whenever the club administrator added or removed a team member from the team which would be frustrating to the user. The author utilised AJAX which allowed the author to implement code that can update the team members on the team without refreshing the full page.



```
function addMember(teamID, userID) {
    // URL for add team member controller method
    var url = "/teams/addTeamMember/" + teamID + "/" + userID;
    //Loads fragment table with updated table of team members
    $('#teamMembersTable_div').load(url);
}

function removeMember(teamID, userID) {
    //URL for remove team member controller method
    var url = "/teams/removeTeamMember/" + teamID
    + "/" + userID;
    //Loads fragment table with updated table of team members
    $('#teamMembersTable_div').load(url);
}
```

Figure 27: Add Team Member and Remove Team Member Functions using AJAX

In the figure above, the author implemented two functions for adding and removing a team member from a team. The add member method makes a request to the controller endpoint which adds a new member to the team and the remove member method makes a request to another controller endpoint which removes a member from the team. It then loads a fragment into the “#teamMembersTable_div” which consists of the updated “Current Team Members” and “Available Players” tables without refreshing the whole web page.

4.9 Implementing Graphs Using D3

The author wanted to apply descriptive analytics to his application which consists of using data visualisations techniques to describe current and historical data which can help to identify trends or patterns in data. Implementing suitable graphs that are interactive so that users can easily analyse and discover trends in the sports club was one of the main priorities of this project. Examples of these graphs include a bar chart that displays the current amount of male and female members in the sports club and a line graph that can display a player’s height, weight, body fat percentage and BMI trends over time. The author used D3 to implement these graphs.

Members Distribution by Gender Bar Chart

On the club member statistics web page, the distribution of members by gender bar chart was implemented to show the number of males and females that are currently members in the sports club based on the role selected by the club administrator. The options included “All Members”, “Club Members”, “Trainers”, “Players” and “Admins”. When the club admin selects a different option, the bars on the chart updates through a short transition according to the data.

In order for the data to be used in the graphs, they needed to be parsed into JSON. The author used the Jackson Library to convert Java objects to JSON String which can then be parsed into JSON.



```
clubAdminGenderCount.add(Map.of("gender", "Male", "value", allMaleAdmin));
clubAdminGenderCount.add(Map.of("gender", "Female", "value", allFemaleAdmin));
String totalClubAdmins = objectMapper.writeValueAsString(clubAdminGenderCount);
```

Figure 28: Converting a List Of Maps to JSON String using the Jackson Library

In the figure above, the “clubAdminGenderCount” list adds two maps. One map consists of the number of male club administrators and the other map consists of the number of female club administrators recorded on the sport club management system. The Object Mapper instance “objectMapper” calls the “writeValueAsString” method which converts the “clubAdminGenderCount” list to a JSON string.

```
[{"value":1,"gender":"Male"}, {"value":0,"gender":"Female"}]
```

Figure 29: Output of Club Admin Gender Count List in JSON string Format



```
<script th:inline="javascript">

    window.onload = function() {
        drawGenderChart([[${totalClubUsers}]],'All Members','Gender');

        displayAgeGroupBarChart([[${totalClubUsersAgeGroup}]]);

        let usersRegisteredData=
JSON.parse([[${listOfNewUsersPast6monthsJson}]])
        displayUserRegisteredLineChart(usersRegisteredData);

    };
</script>
```

Figure 30: JavaScript Functions Being Called To Draw Graphs.

The “drawGenderChart”, “displayAgeGroupBarChart” and the “displayUserRegisteredLineChart” functions are called when the club member statistics HTML page is loaded.

```

function drawGenderChart(data, yAxisLabel) {
    // Data is parsed into JSON format
    data= JSON.parse(data);

    var maxValue = d3.max(data, (d)=> { return d.value; });

    //Setting the dimensions for the bar graph
    margin = { top: 60, right: 30, bottom: 70, left: 60 };
    width = 600 - margin.left - margin.right;
    height = 400 - margin.top - margin.bottom;

    // Adding an SVG element to gender barchart div
    svg = d3.select("#gender-barchart")
        .append("svg")
        .attr("width", width + margin.left + margin.right)
        .attr("height", height + margin.top + margin.bottom)
        .append("g")
        .attr("transform", "translate(" + margin.left + "," + margin.top +
    ")");
    // Creating the band scale for the x-axis
    x = d3.scaleBand()
        .range([0, width])
        .domain(data.map((d)=> { return d.gender; }))
        .padding(0.2);
    // Adding the x-axis to the bar graph.
    svg.append("g")
        .attr("transform", "translate(0," + height + ")")
        .call(d3.axisBottom(x));

    // Creating the linear scale for the y-axis
    y = d3.scaleLinear()
        .domain([0, maxValue])
        .range([height, 0]);
    // Adding the y-Axis to the bar graph
    yAxisGroup=svg.append("g")
        .call(d3.axisLeft(y));

    //Adding the chart title to the graph
    svg.append("text")
        .attr("x", width / 2)
        .attr("y", -20)
        .attr("text-anchor", "middle")
        .style("font-size", "16px")
        .style("font-weight", "bold")
        .text("Members Distribution by Gender");

    // Adding the x-axis label to the graph
    xAxisTitle = svg.append("text")
        .attr("x", width / 2)
        .attr("y", height + 40)
        .attr("text-anchor", "middle")
        .style("font-size", "14px")
        .text("Gender");

    // Adding the y-axis label to the graph
    yAxisTitle = svg.append("text")
        .attr("transform", "rotate(-90)")
        .attr("x", -height / 2)
        .attr("y", -50)
        .attr("text-anchor", "middle")
        .style("font-size", "14px")

    // Calling the "updateGenderChart" function to draw the bars on the graph.
    updateGenderChart(data, yAxisLabel);
}

```

Figure 31: Draw Gender Chart Function

The “draw gender chart” initializes all the components for the member distribution by Gender Bar Chart. The data is parsed into JSON Format. The margin, width, height dimensions for the graph are then initialized. An SVG element is placed inside the “gender-barchart” div that is defined in the club member statistics HTML file. The “d3.scaleBand()” divides the range which is the width of the bar graph into two equal bands for the male and female genders. The “d3.axisBottom (x)” function creates the x-axis based on the band scale that was created and it is added to the graph. The “d3.scaleLinear()” function creates a linear scale that maps the input values which is the values of the data passed into the function to the output values based on the height of the graph. The left vertical axis is created using the “d3.axisLeft(y)” function based on the created linear scale, the Y-axis is then added to the graph. The x-axis label, y-axis label and the chart title is added to the graph. It then calls the “updateGenderChart(data, xAxisLabel, yAxisLabel)” which will draw the bars on the bar chart.

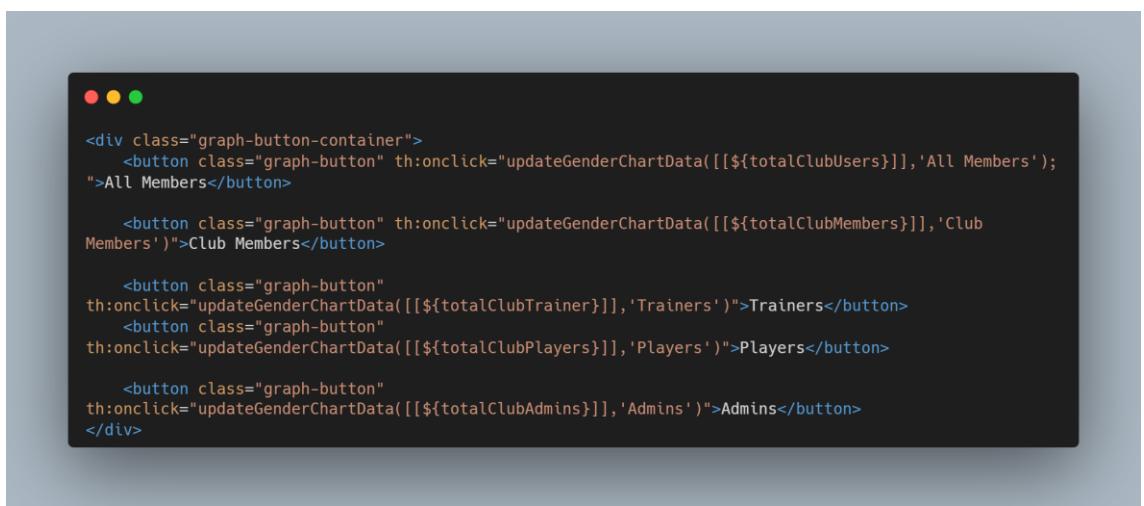
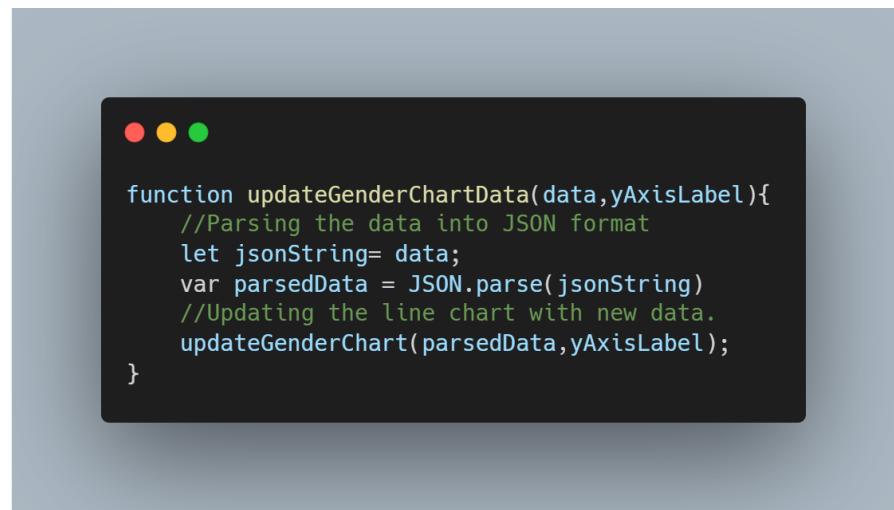


Figure 32: Buttons That Update the Bar Chart Depending on the Role Selected.

The club administrator can select a button to view how many males and females in the selected role are recorded on the sport club management system. This calls the “updateGenderChartData” function which accepts the data and the y-axis label as parameters.



```
function updateGenderChartData(data,yAxisLabel){  
    //Parsing the data into JSON format  
    let jsonString= data;  
    var parsedData = JSON.parse(jsonString)  
    //Updating the line chart with new data.  
    updateGenderChart(parsedData,yAxisLabel);  
}
```

Figure 33: Update Gender Chart Data Function

The “updateGenderChartData” function parses the data which is in JSON string format to JSON. It then calls the “updateGenderChart” function that will draw the bars on the bar chart according to the data passed in.

```

function updateGenderChart(data, yAxisLabel) {

    var maxValue = d3.max(data, (d)=> { return d.value; });

    // Updating the y-axis domain of the bar graph
    y.domain([0,maxValue])

    // Updating the values on the bar graph's y-axis scale
    yAxisGroup.transition()
        .duration(1000)
        .call(d3.axisLeft(y));

    // Binding the data to the SVG rectangle elements
    var bars =svg.selectAll("rect")
        .data(data);

    // Drawing bars according to the data passed in
    bars.join("rect")
        .on("mouseover", (event, d) => {
            tooltip.style("opacity", 1)
                .html(`<strong>Gender: ${d.gender}</strong><br>New Members Total:
${d.value}`)    .style("left", (event.pageX + 15) + "px")
                .style("top", (event.pageY + 15) + "px");
        })
        .on("mouseout", () => tooltip.style("opacity", 0))
        .transition()
        .duration(1000)
        .attr("x", (d) => { return x(d.gender); })
        .attr("y", (d) => { return y(d.value); })
        .attr("width", x.bandwidth())
        .attr("height", (d) => { return height - y(d.value); })
        .attr("fill", "#1f77b4")

    // Updating the y-axis title on the bar graph.
    yAxisTitle.text(yAxisLabel);

}

```

Figure 34: Update Gender Chart Function

The update gender chart function draws the bars on the bar chart which displays the number of male and female members on the bar chart. It selects all of the “rect” svg elements which are the bars of the graph and binds the data to them. The “join” method is responsible for appending, reordering and removing any rectangle elements in order to match the data that is passed into the function. The shape of the bars are then created based on the data values for the number of males and female members recorded on the system. A tooltip is also applied to the graph which is a little textbox that appears when a user hovers over a bar in which it displays the value.

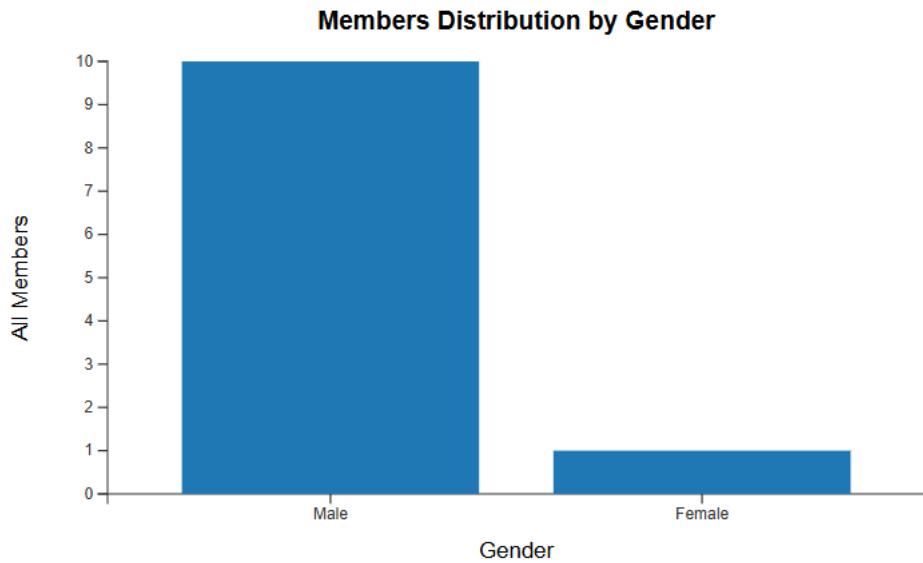


Figure 35: Members Distribution Bar Chart Displaying the Number of Male and Female Members

Player Physical Stats Line Chart

The author implemented a line chart to display a player's physical stats which consisted of their weight(kg), height(m), body fat percentage and their BMI which is calculated by the following formula ($\text{player weight} / \text{player height}^2$). The author selected these attributes as afrom the "Gym Members Exercise Dataset" dataset by Vala Khorasani on Kaggle (Khorasani, 2024). Like the member distribution by gender bar chart, the trainer can select to view the player's height, weight, body fat percentage and BMI trends over a period of time.

```

function drawPlayerPhysicalStatsLineChart(stringData, yAxisText, playerName ) {
    //Parsing the data to JSON format
    data = JSON.parse(stringData, (key, value) => {
        if (key === "date") {
            return new Date(value);
        }
        return value;
    });
    margin = {top: 60, right: 50, bottom: 50, left: 60},
    width = 800 - margin.left - margin.right,
    height = 400 - margin.top - margin.bottom;

    // Appending an SVG element to the physical-stats-line-chart div
    svg = d3.select("#physical-stats-line-chart")
    .append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
    .append("g")
    .attr("transform", `translate(${margin.left},${margin.top})`);

    // Creating the x and y-axis scales for the line graph
    x = d3.scaleTime().range([0, width]);
    y = d3.scaleLinear().range([height, 0]);

    // Appending groups for x and y axes on the line graph
    xAxisGroup = svg.append("g")
    .attr("transform", `translate(0,${height})`);
    yAxisGroup = svg.append("g");

    // Appending a group element for the line on the line graph
    linePathGroup = svg.append("g")

    // Appending the SVG path element for the line
    path = linePathGroup.append("g")
        .append("path")
        .attr("fill", "none")
        .attr("stroke", "steelblue")
        .attr("stroke-width", 2);

    // Creating group for the circles on the line graph
    circlesGroup = svg.append("g");

    // Adding the x-axis label to the line graph
    xAxisLabel = svg.append("text")
        .attr("x", width / 2)
        .attr("y", height + 40)
        .attr("text-anchor", "middle")
        .style("font-size", "14px")
        .text("Date Recorded");

    // Adding the y-axis label to the line graph
    yAxisLabel = svg.append("text")
        .attr("transform", "rotate(-90)")
        .attr("x", -height / 2)
        .attr("y", -50)
        .attr("text-anchor", "middle")
        .style("font-size", "14px");

    // Adding the chart title to the line graph
    chartTitle= svg.append("text")
        .attr("x", width / 2)
        .attr("y", -20)
        .attr("text-anchor", "middle")
        .style("font-size", "18px")
        .style("font-weight", "bold");

    updateLineChart(data, yAxisText, playerName);
}

```

Figure 36: Draw Player Physical Stats Line Chart Function

The “drawPlayerPhysicalStatsLineChart” configures all of the main components of the line graph. The margin, width and height dimensions of the graph are defined. The variables for the creation of the line path, circles, x and y axis of the line graph, x and y axis labels and the chart title of the line graph. The “updateLineChart” function is then called which draws the line and the circles on the line graph according to the data passed in.



```
// Function to draw the line and circles of the line graph according to the data passed in
function updateLineChart(data, yAxisText, playerName) {

    // Updating the x and y-axis domains on the line graph according to the data passed in
    x.domain(d3.extent(data, d => d.date));
    y.domain([0, d3.max(data, d => d.value)]).nice();

    const tickDates = data.map(d => d.date);

    const tooltipTimeFormat=d3.timeFormat("%d %b %Y");

    // Updating the X and Y axes on the line graph according to the data passed in
    xAxisGroup.transition().duration(1000)
        .call(d3.axisBottom(x).tickValues(tickDates).tickFormat(d3.timeFormat("%d-%m-%Y")));
    yAxisGroup.transition().duration(1000)
        .call(d3.axisLeft(y));

    // Creating a function that returns an SVG path element which describes the shape of the line
    line = d3.line()
        .x((d) => { return x(d.date); })
        .y((d) => { return y(d.value); });

    // Updating the line on the line graph according to the line generated based on the data.
    path.datum(data)
        .transition()
        .duration(1000)
        .attr("d", line);

    // Binding the data to all of the circle elements on the line graph.
    var circles = circlesGroup.selectAll("circle")
        .data(data);

    // Updating the circles positions on the line graph according to the data passed in
    circles.join("circle")
        .attr("cx", (d)=> { return x(d.date); })
        .attr("cy", (d)=> { return y(d.value); })
        .attr("r", 0)
        .attr("fill", "red")
        .attr("r", 4)
        .on("mouseover", (event, d) => {
            lineGraphTooltip.style("opacity", 1)
                .html(`<strong>Date: </strong>${tooltipTimeFormat(d.date)}<br><strong>Value:</strong> ${d.value}`);
            .style("left", (event.pageX + 15) + "px")
            .style("top", (event.pageY + 15) + "px");
        })
        .on("mouseout", () => lineGraphTooltip.style("opacity", 0))
        .transition()
        .duration(1000);
    //Updating the chart title and the y-axis label on the line graph.
    chartTitle.text(playerName+ " "+yAxisText+" Trends Over Time");
    yAxisLabel.text(yAxisText);
}

}
```

Figure 37:Update Line Chart Function

The “updateLineChart” function is responsible for drawing the shape of the line and the position of the circles on the line graph according to the data passed in. When the data is passed into the function, the x and y axis of the graph are updated according to the data passed into the function. The “d3.line()” creates a function which returns an SVG path element the describes the shape of the line according to the data passed into the function. The line variable which contains the “d3.line()” function that defines the shape of the line is then applied to the path’s “d” attribute which draws the line. The circles are then positioned onto the graph according to the data that is passed into the function. The tool tip feature is also applied which allows the user to display the player physical stats value for a given date when they hover over a circle. The y axis label of the graph is updated based on the “yAxisText” variable that is passed into the function.

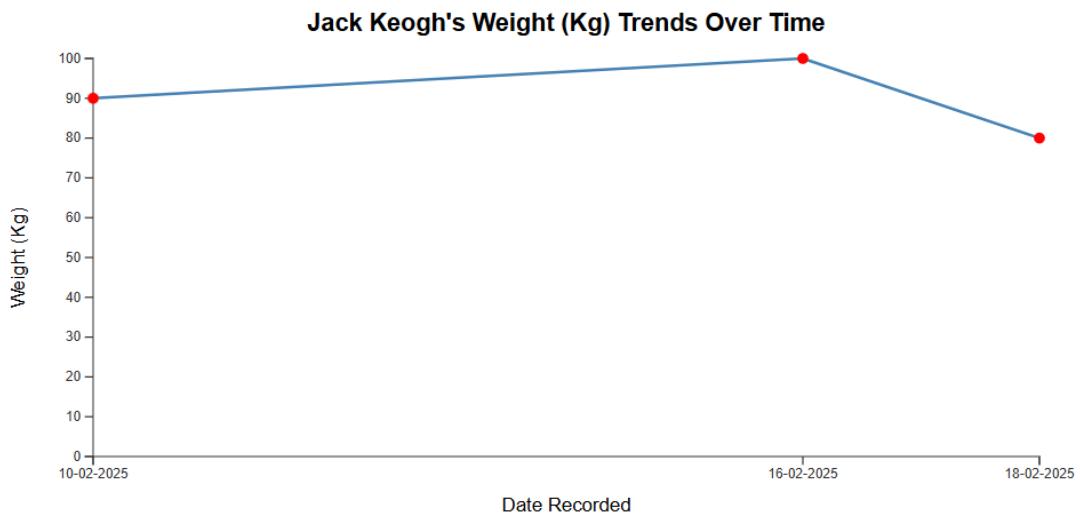
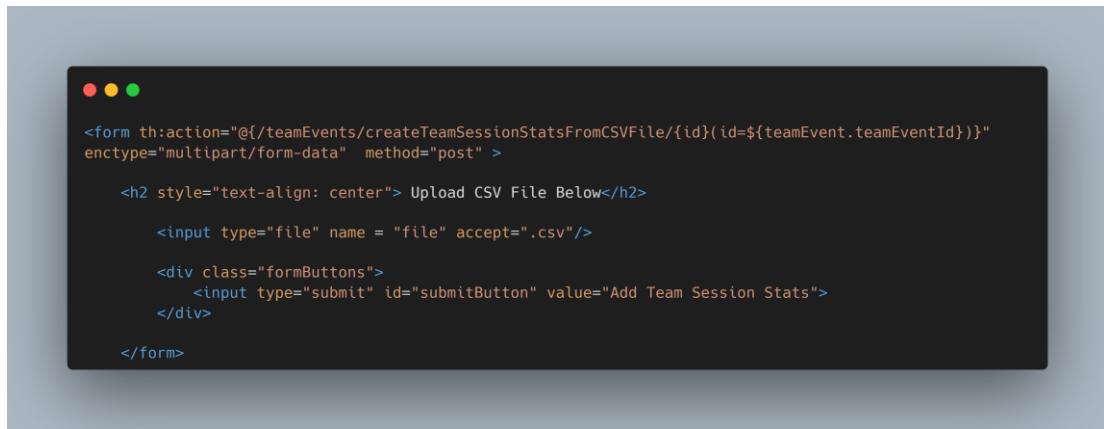


Figure 38: Line Chart Showing Jack Keogh's Weight Trends Over Time

4.10 Uploading Team Session Statistics CSV File

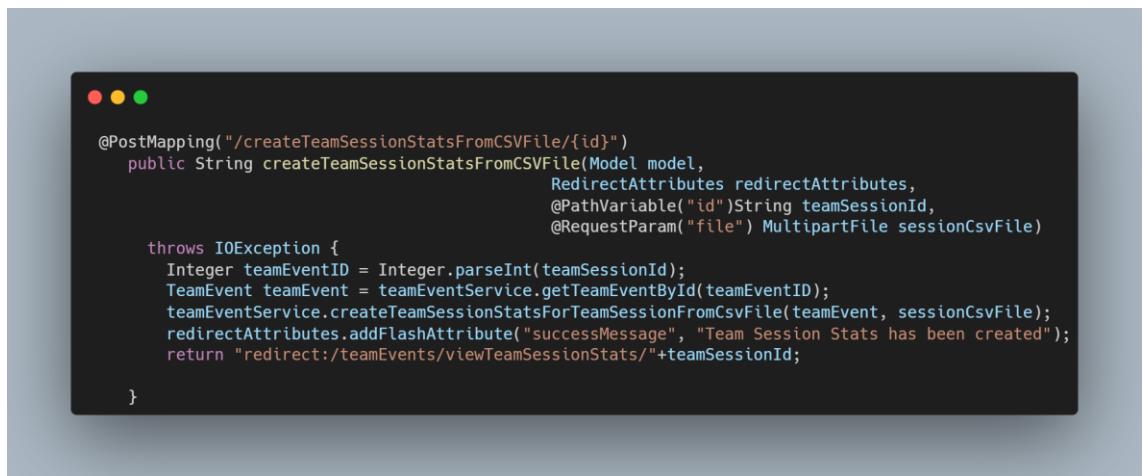
One of the other main challenges in this project was to be able to retrieve the data from a team training session in a suitable manner. Since it is possible to export data recorded from an Apex GPS Device. The author decided to implement a feature that allowed a trainer to upload a CSV file which contained the statistics recorded in a team session to the sports club management application as a proof of concept. To simulate the type of data that could be recorded on each player in a team session, the author selected the max BPM, resting BPM, average BPM and calories burned attributes from the “Gym Members Exercise Dataset” dataset by Vala Khorasani on Kaggle (Khorasani, 2024). The author used the Apache Commons CSV library to implement this feature.



```
<form th:action="@{/teamEvents/createTeamSessionStatsFromCSVFile/{id}(id=${teamEvent.teamEventId})}"  
enctype="multipart/form-data" method="post">  
  
    <h2 style="text-align: center"> Upload CSV File Below</h2>  
  
    <input type="file" name = "file" accept=".csv"/>  
  
    <div class="formButtons">  
        <input type="submit" id="submitButton" value="Add Team Session Stats">  
    </div>  
  
</form>
```

Figure 39:Upload Team Session Stats CSV File HTML Form Code

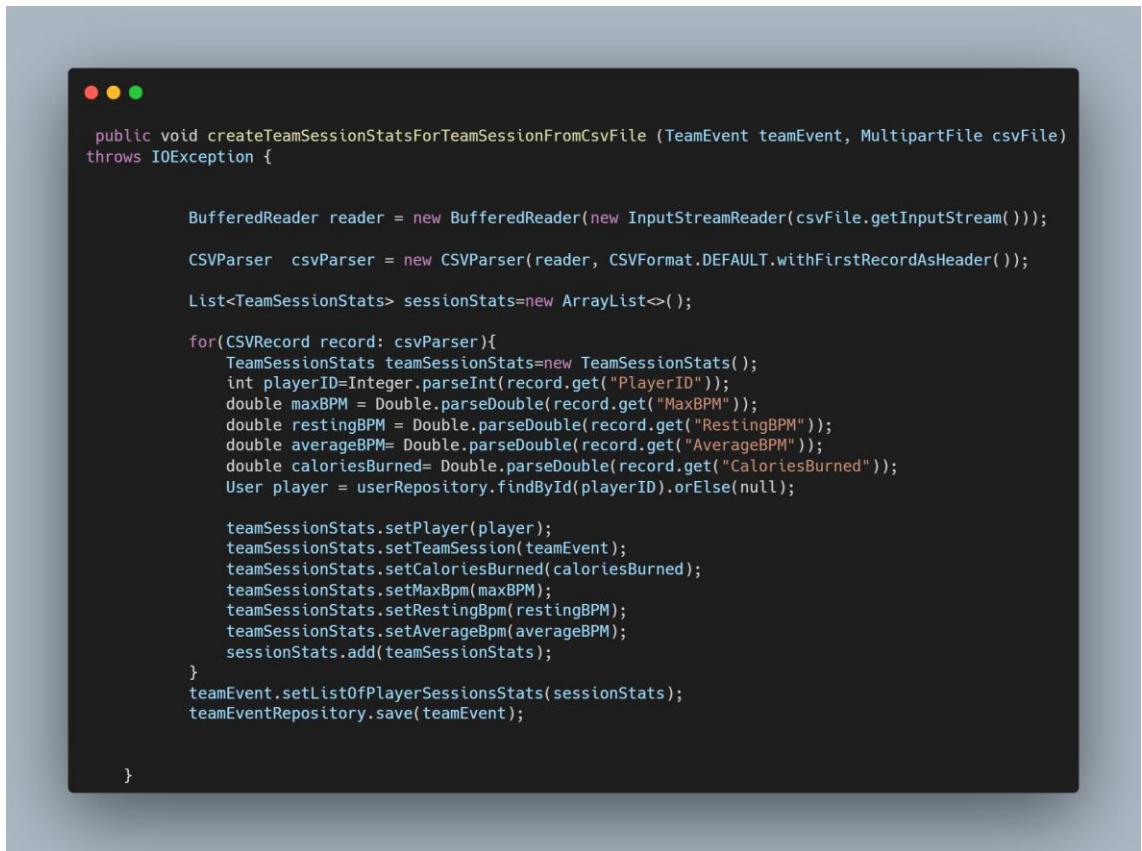
This form enables the trainer to upload a CSV file and record the team session statistics to the Sports Club Management system. When the trainer presses the submit button, it sends the submitted file to a URL endpoint in the “teamEventController” controller class through a POST request.



```
@PostMapping( "/createTeamSessionStatsFromCSVFile/{id}" )  
public String createTeamSessionStatsFromCSVFile(Model model,  
                                              RedirectAttributes redirectAttributes,  
                                              @PathVariable("id")String teamSessionId,  
                                              @RequestParam("file") MultipartFile sessionCsvFile)  
throws IOException {  
    Integer teamEventID = Integer.parseInt(teamSessionId);  
    TeamEvent teamEvent = teamEventService.getTeamEventById(teamEventID);  
    teamEventService.createTeamSessionStatsForTeamSessionFromCsvFile(teamEvent, sessionCsvFile);  
    redirectAttributes.addFlashAttribute("successMessage", "Team Session Stats has been created");  
    return "redirect:/teamEvents/viewTeamSessionStats/"+teamSessionId;  
}
```

Figure 40: Create Team Session Stats From CSV File Controller Method

When it receives the POST request, the “createTeamSessionStatsFromCSVFile” controller method is invoked. The team event ID is passed in through the URL path, the team event ID is passed into the “getTeamEventById” service method to retrieve the team event object. The “createTeamSessionStatsForTeamSessionFromCSVFile” service method in the “TeamEventService” class accepts the team event object and the CSV file to upload the team session stats to the system. It redirects the user back to the view team session stats page and displays a message stating that the “Team Session Stats has been created”.



```
public void createTeamSessionStatsForTeamSessionFromCsvFile (TeamEvent teamEvent, MultipartFile csvFile)
throws IOException {

    BufferedReader reader = new BufferedReader(new InputStreamReader(csvFile.getInputStream()));

    CSVParser csvParser = new CSVParser(reader, CSVFormat.DEFAULT.withFirstRecordAsHeader());

    List<TeamSessionStats> sessionStats=new ArrayList<>();

    for(CSVRecord record: csvParser){
        TeamSessionStats teamSessionStats=new TeamSessionStats();
        int playerID=Integer.parseInt(record.get("PlayerID"));
        double maxBPM = Double.parseDouble(record.get("MaxBPM"));
        double restingBPM = Double.parseDouble(record.get("RestingBPM"));
        double averageBPM= Double.parseDouble(record.get("AverageBPM"));
        double caloriesBurned= Double.parseDouble(record.get("CaloriesBurned"));
        User player = userRepository.findById(playerID).orElse(null);

        teamSessionStats.setPlayer(player);
        teamSessionStats.setTeamSession(teamEvent);
        teamSessionStats.setCaloriesBurned(caloriesBurned);
        teamSessionStats.setMaxBpm(maxBPM);
        teamSessionStats.setRestingBpm(restingBPM);
        teamSessionStats.setAverageBpm(averageBPM);
        sessionStats.add(teamSessionStats);
    }
    teamEvent.setListOfPlayerSessionsStats(sessionStats);
    teamEventRepository.save(teamEvent);

}
```

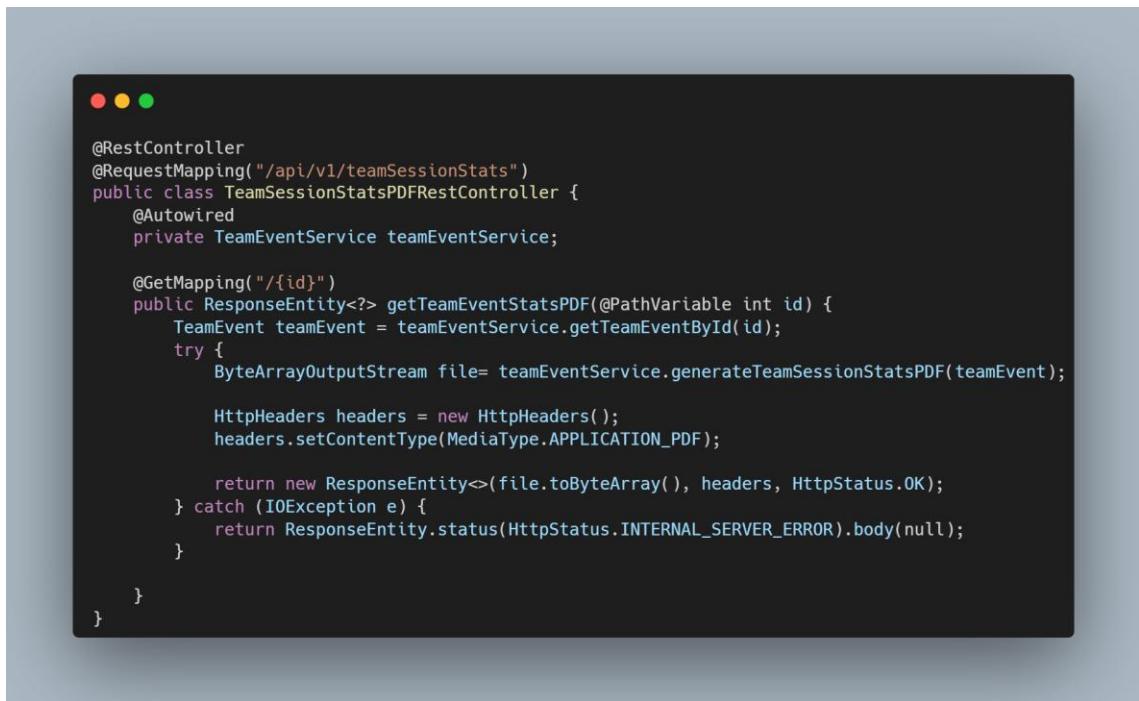
Figure 41: Create Team Session Stats For Team Session From CSV File Service Method

When this method is invoked, the Buffered Reader reads the text from the CSV file character-input stream. The “CSVParser” parses the CSV file to the specified format and returns a “CSVRecord” object for each row in the CSV file. This allowed the author to create “TeamSessionStats” objects by accessing each “CSVRecord” object generated by the “CSVParser” class. Each team session stats object is added to a list and set to the team event object. The team event object is updated on the database. The team session statistics is also saved to the database since it has a relationship with the team event object.

4.11 REST API To Export Team Session Statistics to a PDF File.

This feature enables trainers to export statistics recorded from a selected team session to a PDF file. This could allow the trainer to print out the team session statistics if they wanted to present it to someone else in person. The author developed a small REST API that would generate a PDF file with the recorded team session statistics in a table format. The author attended Alan Ryan’s lecture and tutorials in the “API Design and

Development” module which is where the author learned how to develop REST APIs in Spring Boot. To create the PDF file that contained all of the team session statistics, the author used the iText Core library.

A screenshot of a code editor window showing Java code. The code defines a REST controller for handling team session statistics. It includes annotations for @RestController and @RequestMapping, and uses @Autowired to inject a TeamEventService. The getTeamEventStatsPDF method retrieves a team event by ID and generates a PDF response with the correct content type.

```
@RestController
@RequestMapping("/api/v1/teamSessionStats")
public class TeamSessionStatsPDFRestController {
    @Autowired
    private TeamEventService teamEventService;

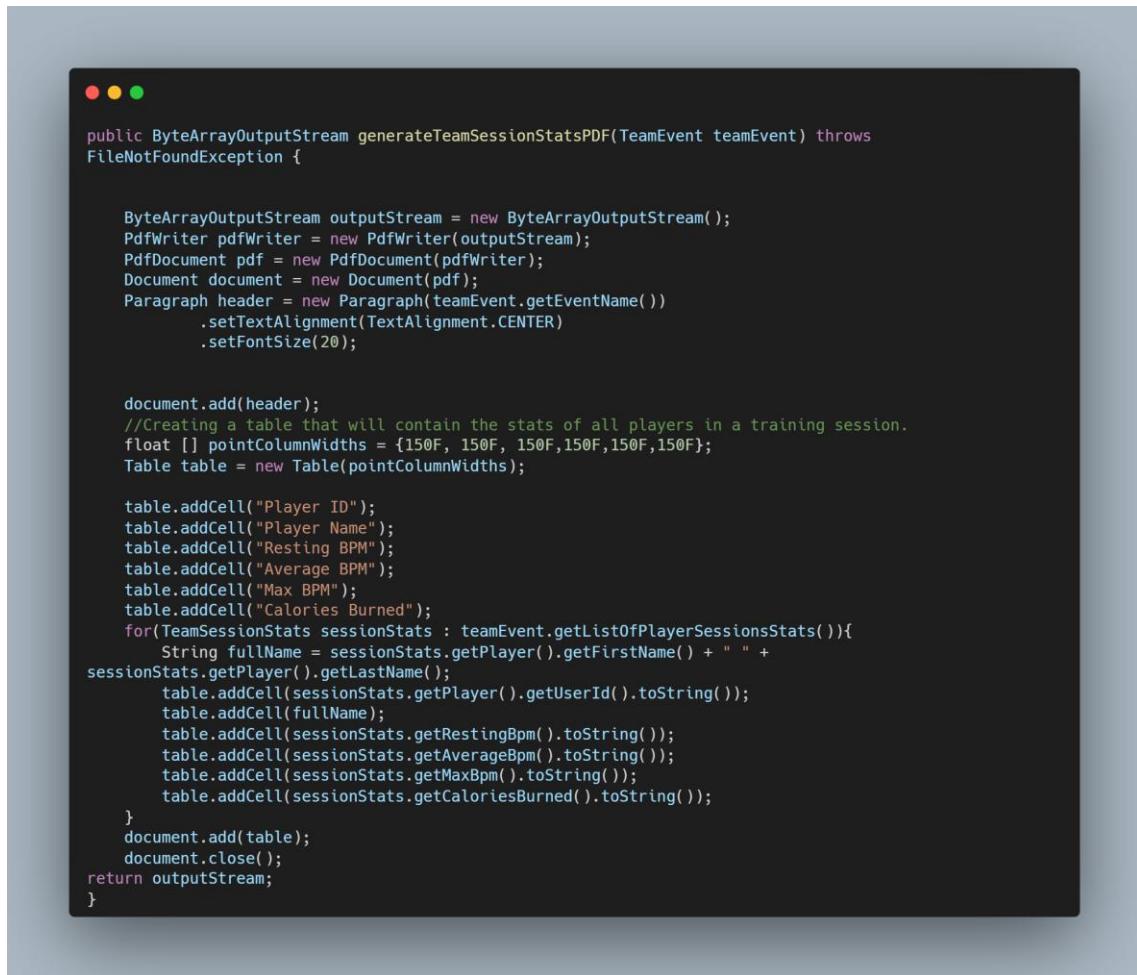
    @GetMapping("/{id}")
    public ResponseEntity<?> getTeamEventStatsPDF(@PathVariable int id) {
        TeamEvent teamEvent = teamEventService.getTeamEventById(id);
        try {
            ByteArrayOutputStream file= teamEventService.generateTeamSessionStatsPDF(teamEvent);

            HttpHeaders headers = new HttpHeaders();
            headers.setContentType(MediaType.APPLICATION_PDF);

            return new ResponseEntity<>(file.toByteArray(), headers, HttpStatus.OK);
        } catch (IOException e) {
            return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body(null);
        }
    }
}
```

Figure 42: Team Session Stats PDF REST Controller Method

The “@RestController” annotation allows the class to handle REST requests such as POST, GET, DELETE, and PUT. When a GET request is made to the REST API, it retrieves the team session id from the URL path and retrieves the team event object from the “getTeamEventById” method in the “TeamEventService” class. It then passes the team event object to the “generateTeamSessionStatsPDF” method which creates the contents of the PDF file. The content type is set to “MediaType.APPLICATION_PDF” which states that the content is a PDF file and is included in the HTTP header. The “ResponseEntity” object represents the HTTP response which includes the PDF File as the body, HTTP header and the response status code of 200 to indicate that the request has been completed.



```
public ByteArrayOutputStream generateTeamSessionStatsPDF(TeamEvent teamEvent) throws FileNotFoundException {
    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
    PdfWriter pdfWriter = new PdfWriter(outputStream);
    PdfDocument pdf = new PdfDocument(pdfWriter);
    Document document = new Document(pdf);
    Paragraph header = new Paragraph(teamEvent.getEventName())
        .set.TextAlignment(TextAlignment.CENTER)
        .setFontSize(20);

    document.add(header);
    //Creating a table that will contain the stats of all players in a training session.
    float [] pointColumnWidths = {150F, 150F, 150F,150F,150F,150F};
    Table table = new Table(pointColumnWidths);

    table.addCell("Player ID");
    table.addCell("Player Name");
    table.addCell("Resting BPM");
    table.addCell("Average BPM");
    table.addCell("Max BPM");
    table.addCell("Calories Burned");
    for(TeamSessionStats sessionStats : teamEvent.getListOfPlayerSessionsStats()){
        String fullName = sessionStats.getPlayer().getFirstName() + " " +
        sessionStats.getPlayer().getLastName();
        table.addCell(sessionStats.getPlayer().getUserId().toString());
        table.addCell(fullName);
        table.addCell(sessionStats.getRestingBpm().toString());
        table.addCell(sessionStats.getAverageBpm().toString());
        table.addCell(sessionStats.getMaxBpm().toString());
        table.addCell(sessionStats.getCaloriesBurned().toString());
    }
    document.add(table);
    document.close();
    return outputStream;
}
```

Figure 43: Generate Team Session Stats PDF Service Method

In this method, the contents of the PDF file is created. The “ByteArrayOutputStream” creates an output stream which writes the data into a byte array. The “PdfWriter” class writes the PDF file to the output stream. The “PdfDocument” represents the PDF document and accepts the “PdfWriter” instance as an argument. The “Document” class accepts the “PdfDocument” instance as an argument, the “Document” class allowed the author to style the PDF document by adding a table and paragraphs to display the team session statistics in an easy-to-read format. The author then added a paragraph which displayed the name of the team session and then created a table that contained the statistics recorded from the team session. The table is then added to the document and then closed. The function then returns the output stream which contains the PDF file.

5 Testing And Results

During the implementation process of the project, the author tested each new feature that was implemented to ensure that it was meeting the specified requirements and remove any bugs that the author encountered. This is also an important stage of the project as the application should be able to operate at an acceptable level for all users. The author used functional testing to monitor the applications behaviour as it was more convenient for him to test all of the features of the application.

5.1 Functional Testing

Functional testing is a black box testing method that is used to test features of a software system to ensure that it behaves correctly according to the requirements and specifications. Black box testing is a software testing method in which the tester is more concerned about the features producing the correct output rather than what goes in the internal code structure. The author used functional testing to test the application by selecting suitable test inputs according to certain features to ensure that they are behaving correctly.

5.2 Test Cases

The author tested all features of the application to ensure that it behaves correctly. The author documented the following test cases as the author believed that they were the most important use cases of the application.

5.2.1 Add A Club Event

This use case allowed club administrators to create a new club event for the sports club. The author selected the following inputs to test the “Add A Club Event” use case.

Test Inputs:

- **Event Name:** Hurling Final
- **Event Description:** Hurling Final against Clonlara
- **Start Date:** 17/07/2025 15:00
- **Location:** Cusack Park, Ennis

Sport Club Club Events Teams View Club Member Stats Register A Club Member Log Out

Add A Club Event

Event Name: Hurling Final

Event Description: Hurling final against Clonlara

Start Date: 17/07/2025 15:00

Location: Cusack Park, Ennis

Add Club Event

Figure 44: Add Club Event Test Data

Result:

The system directs the user to the view club events page. The “Hurling Final” club event has been successfully added to the system and a success message is displayed to the screen.

Sport Club Club Events Teams View Club Member Stats Register A Club Member Log Out

Club Events

Club Event has been created

Add A Club Event

Event Name	Event Location	Start Date And Time	Going	Not Going	Edit	Remove
dinner dance	Castle Oaks Hotel	2025-08-22 20:30	Going	Not Going	Edit	Remove
Hurling Final	Cusack Park, Ennis	2025-07-17 15:00	Going	Not Going	Edit	Remove

Figure 45: Add A Club Event Output

5.2.2 Log Club Event Attendance

This use case allows all users of the application to log if they are going or not going to a club event. The author logged in as “John Doe” and selected to go to the “Hurling Final” event.

Test Input:

- Logged in as John Doe
- Selected “Going” to “Hurling Final” club event.

Result:

The system displays a success message to the screen and John Doe's attendance to the "Hurling Final" is recorded to the system.

Club Events				
Club event attendance recorded				
Event Name	Event Location	Start Date And Time	Going	Not Going
dinner dance	Castle Oaks Hotel	2025-08-22 20:30	Going	Not Going
Hurling Final	Cusack Park, Ennis	2025-07-17 15:00	Going	Not Going

Figure 46: The Log Event Attendance Output

John Doe's attendance for the hurling final has been recorded to the system.

Hurling Final Details	
Event Description	Hurling final against Clonlara
Event Location	Cusack Park, Ennis
Event Start Date	2025-07-17 15:00
Back To Club Events	
Going: 1	
Name	Email
John Doe	JohnDoe@gmail.com
Not Going: 0	
Name	Email

Figure 47: John Doe's Attendance Displayed on The Hurling Final Club Event

5.2.3 Add A Team

This use case allows club administrators to create a new team, select a trainer and add players to the team. The author selected the following test inputs to test the "Add A Team" use case.

Test Inputs:

- **Team Name:** Senior Hurling Team
- **Trainer:** Mary Smith
- **Players:** Jack Keogh, Jake O' Brien, Jason Powell.

Team Name:
Senior Hurling Team

Trainer:
Mary Smith

Team Members: (Hold Ctrl button and select players to add to the team)

Jack Keogh
Jake O'Brien
Jason Powell
Jason Pollock

Add Team

Figure 48: Add A Team Test Data

Result:

The system directs the user to the view teams which includes the “Senior Hurling Team” and displays a success message.

Team has been created

Add A Team

Team Name	Trainer	Edit	Remove
hurling team	Mary Smith	Edit	Remove
senior football team	Mary Smith	Edit	Remove
senior soccer team	Mary Smith	Edit	Remove
Senior Hurling Team	Mary Smith	Edit	Remove

Figure 49: Add A Team Output.

The new senior hurling team has been successfully added to the system.

Senior Hurling Team Details

Trainer : Mary Smith

Team Members

Name	Email
Jack Keogh	JKeogh@gmail.com
Jake O'Brien	JOB@gmail.com
Jason Powell	JasonPowell@Gmail.com

Back To Teams

Figure 50: The New Senior Hurling Team Details.

5.2.4 Edit A Team

This use case allows the club administrator to edit the team's name, select a different trainer and add players or remove current players from a selected team. The author decided to create a new player called "Jason Powell" to add to the "Senior Hurling Team". The author also changed the team's name to "Intermediate Hurling Team" and removed the player "Jack Keogh" from the team.

Test Inputs:

- Add Jason Pollock to the team.
- Change team name to "Intermediate Hurling Team"
- Remove player "Jack Keogh" from the team.

The screenshot shows a web application interface titled 'Edit Team'. At the top, there is a navigation bar with links: Sport Club, Club Events, Teams, View Club Member Stats, Register A Club Member, and Log Out. The main content area has a title 'Edit Team'. It contains two sections: 'Current Team Members' and 'Available Players'. Both sections are represented by tables with columns for Name and Email, and a 'Remove' or 'Add' button.

Name	Email	Action
Jake O'Brien	JOB@gmail.com	Remove
Jason Powell	JasonPowell@gmail.com	Remove
Jason Pollock	JasonPollock@gmail.com	Remove

Name	Email	Action
Jack Keogh	JKeogh@gmail.com	Add

Figure 51: Edit Team Test Data

Result:

The system directs the user to view teams screen that includes the updated team and displays a success message to the screen.

Sport Club	Club Events	Teams	View Club Member Stats	Register A Club Member	Log Out
Teams					
Team has been updated					
Add A Team					
Team Name	Trainer	Edit	Remove		
hurling team	Mary Smith	Edit	Remove		
senior football team	Mary Smith	Edit	Remove		
senior soccer team	Mary Smith	Edit	Remove		
Intermediate Hurling Team	Mary Smith	Edit	Remove		

Figure 52: The Edit Team Output

The team details have been updated successfully on the system.

Sport Club	Club Events	Teams	View Club Member Stats	Register A Club Member	Log Out
Intermediate Hurling Team Details					
Trainer : Mary Smith					
Name	Email				
Jason Pollock	JasonPollock@gmail.com				
Jason Powell	JasonPowell@Gmail.com				
Jake O'Brien	JOB@gmail.com				
Back To Teams					

Figure 53: The Senior Hurling Team Updated Details.

5.2.5 Add A Team Session

This use case allows trainers to add a new team session for their selected team. The author selected the following test inputs.

Test Inputs:

- **Team Session Name:** Hurling training
- **Team Session Description:** Hurling training for the hurling final.
- **Team Session Location:** Killaloe GAA grounds.
- **Start Date:** 18/06/2025 19:00
- **Team:** Intermediate Hurling Team

Sport Club Club Events Teams Team Sessions Log Out

Add Team Session

Team Session Name:
Hurling training

Team Session Description:
Hurling training for the hurling final.

Team Session Location:
Killaloe GAA Grounds

Start Date:
18/06/2025 19:00

Team:
Intermediate Hurling Team

Add Team Session

Figure 54: The Add Team Session Test Data.

Result:

The system directs the user to the view team sessions page with the new team session and displays a success message. The new team event has been successfully added to the system.

Sport Club Club Events Teams Team Sessions Log Out

Team Sessions

Team Session has been created

Add A Team Session

Session	Session Location	Start Date And Time	Team	Edit	Add Team Session Stats	View Team Session Stats	Remove
Hurling training	Killaloe GAA Grounds	2025-06-18 19:00	Intermediate Hurling Team	Edit	Add Team Session Stats	View Team Session Stats	Remove

Figure 55: The Add Team Session Output.

5.2.6 Upload Team Session Statistics CSV File

This use case allows trainers to upload a CSV file that contains statistics from each player in a team session. The author created a sample CSV file that contains statistics recorded from each player in the “Intermediate Hurling Team” during their “Hurling training” team session.

Test Input CSV File:

PlayerID	MaxBPM	RestingBPM	AverageBPM	CaloriesBurned
7	187	176	174	205
9	195	167	186	267
57	201	161	186	300

Figure 56: Hurling Training Team Session Stats CSV File.

Player ID:

7: Jake O'Brien

9: Jason Powell

57: Jason Pollock

Result:

The team session details have been successfully added to the system, the user is redirected to the hurling training team session statistics page which displays the recorded statistics and a success message is displayed to the screen.

The screenshot shows a web application interface for managing team session statistics. At the top, there is a navigation bar with links: Sport Club, Club Events, Teams, Team Sessions, and Log Out. Below the navigation bar, the title "Hurling training Session Stats" is centered. A green success message "Team Session Stats has been created" is displayed above a "Export To PDF" button. Below the message is a table with five columns: Name, Resting BPM, Average BPM, Max BPM, and Calories Burned. The table contains three rows of data corresponding to the players listed in Figure 56: Jake O'Brien (Resting BPM: 176.00, Average BPM: 174.00, Max BPM: 187.00, Calories Burned: 205.00), Jason Powell (Resting BPM: 167.00, Average BPM: 186.00, Max BPM: 195.00, Calories Burned: 267.00), and Jason Pollock (Resting BPM: 161.00, Average BPM: 186.00, Max BPM: 201.00, Calories Burned: 300.00).

Name	Resting BPM	Average BPM	Max BPM	Calories Burned
Jake O'Brien	176.00	174.00	187.00	205.00
Jason Powell	167.00	186.00	195.00	267.00
Jason Pollock	161.00	186.00	201.00	300.00

Figure 57: The New Team Session Statistics Details.

5.2.7 Add Player Physical Stats

This use case allows trainers to add physical stats on players in their selected team. The author selected the player “Jake Pollock” from the “Intermediate Hurling Team”. The author selected the following test inputs:

Test Inputs:

- **Height (Metres):** 1.96
- **Weight (Kg):** 85
- **Body Fat %:** 20

The screenshot shows a web application interface for adding player physical statistics. At the top, there is a navigation bar with links: Sport Club, Club Events, Teams, Team Sessions, and Log Out. Below the navigation bar, the main title is "Add Player Physical Stats". Underneath it, a sub-section title is "Add Jason Pollock Physical Stats". There are three input fields: "Height (Metres)" containing "1.96", "Weight (Kg)" containing "85", and "Body Fat %" containing "20". A green button at the bottom right of the form area is labeled "Add Player Physical Stats".

Figure 58: Add Player Physical Stats Test Input

Result:

The system directs the user to the view player physical statistics page that displays the recorded player physical stats and also displays a success message. The BMI is generated using the following formula (players weight / (players height)²). The entered height was 1.96 metres, and the entered weight was 85 kg. the following formula would be 85 / (1.95)² which would be equal to 22.13.

The screenshot shows the "Jason Pollock Physical Stats" page. At the top, a success message says "Player physical stats has been created". Below that is a table with four columns: Height (Metres), Weight (Kg), Body Fat %, and Body Mass Index (BMI). The table contains the following data:
Height (Metres): 01.96
Weight (Kg): 85.00
Body Fat %: 20.00
Body Mass Index (BMI): 22.13

Height (Metres)	Weight (Kg)	Body Fat %	Body Mass Index (BMI)
01.96	85.00	20.00	22.13

A green button at the bottom left is labeled "Back To Team Details".

Figure 59: Add Player Physical Stats Output

5.2.8 Export Team Session Statistics To A PDF File

A club trainer is able to export statistics recorded in a team session to a PDF file. The author tested this feature by exporting the statistics recorded for the “Hurling Training” Team Session.

Hurling training Session Stats				
Team Session Stats has been created				
Export To PDF				
Name	Resting BPM	Average BPM	Max BPM	Calories Burned
Jake O'Brien	176.00	174.00	187.00	205.00
Jason Powell	167.00	186.00	195.00	267.00
Jason Pollock	161.00	186.00	201.00	300.00

Figure 60: Hurling Training Recorded Team Session Statistics.

Result:

A PDF file is generated which contains a table of all of the recorded statistics from each player that participated in the “Hurling training” team session.

Hurling training					
Player ID	Player Name	Resting BPM	Average BPM	Max BPM	Calories Burned
7	Jake O'Brien	176.0	174.0	187.0	205.0
9	Jason Powell	167.0	186.0	195.0	267.0
57	Jason Pollock	161.0	186.0	201.0	300.0

Figure 61: Generated PDF File Containing The Hurling Training Team Session Statistics

5.2.9 Members Distribution By Gender Bar Chart

The “Member Distribution By Gender” bar chart on the club member statistics page can be viewed by a club administrator which displays the number of all males and female members registered on the sports club management system. The club administrator can also view how many male and female club members, trainers, players and club administrators are recorded on the system. The author tested the bar chart by selecting the “All Members” option which is all of the members whether they are normal club members, players, trainers or club administrators. The author also selected the “Players” option. During the time of testing the bar chart, there was 24 male members, 7 female members, 19 male players and 0 female players.

Test Data:

Number of Male Members: 24

Number of Female Members: 7

Number of Male Players: 19

Number of Female Players: 0

Result:

The barchart displays 24 male members and 7 female members when the author selected the “All Members” option. The barchart then displayed 19 male players and 0 female players when the author selected the “Players” option.

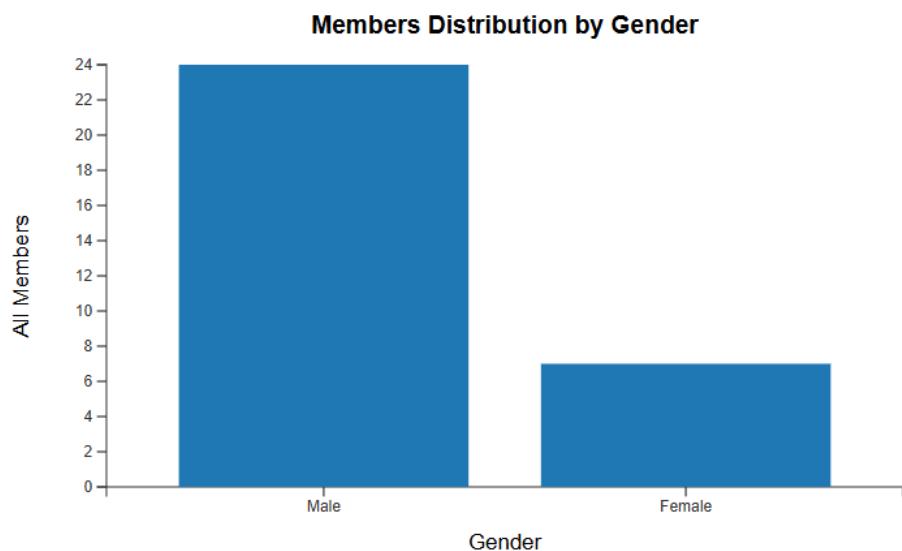


Figure 62: Bar chart displaying the number of Male and Female Members.

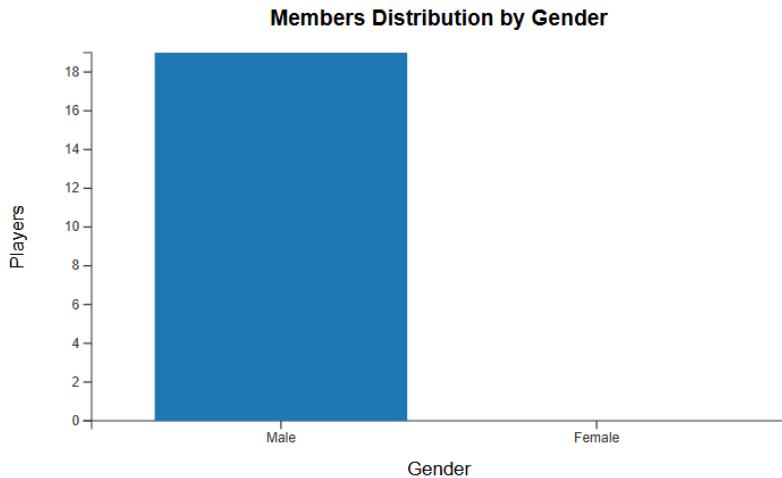


Figure 63: Bar Chart Displaying the Number of Male and Female Players

5.2.10 Player Physical Stats Line Chart

The player physical stats line chart is displayed on the “view player physical statistics” page which can be viewed by a club trainer. The club trainer can view trends in the players weight, height, body fat percentage and body mass index by selecting these options above the graph. The author tested this use case by selecting Jack Keogh’s “Weight” and “Body Fat Percentage” statistics to be displayed on the bar chart. During the time of testing this use case, the players data consisted of the following:

Jack Keoghs Weight Test Data:

Date Recorded: 10-02-2025 **Weight(Kg):** 90

Date Recorded: 16-02-2025 **Weight(Kg):** 100

Date Recorded: 18-02-2025 **Weight(Kg):** 80

Jack Keoghs Body Fat Percentage Test Data:

Date Recorded: 10-02-2025 **Body Fat Percentage:** 18

Date Recorded: 16-02-2025 **Body Fat Percentage:** 20

Date Recorded: 18-02-2025 **Body Fat Percentage:** 15

Sport Club	Club Events	Teams	Team Sessions	Log Out
Jack Keogh Physical Stats				
Height (Metres)	Weight (Kg)	Body Fat %	Body Mass Index (BMI)	
01.80	80.00	15.00	24.69	
02.00	100.00	20.00	25.00	
02.30	90.00	18.00	17.01	

Figure 64: Jack Keogh's Recorded Physical Stats

Result:

The line chart displays Jack Keoghs weight value of “90” that was recorded on the “10-02-2025”, weight value of “100” that was recorded on the “16-02-2025” and weight value of “80” recorded on the “18-02-2025” when the author selected the weight option. The author then selected the “Body Fat Percentage” option which displayed Jack Keogh’s body fat percentage value of “18” that was recorded on the “10-02-2025”, “20” that was recorded on the “16-02-2025”, “15” that was recorded on the “18-02-2025”.

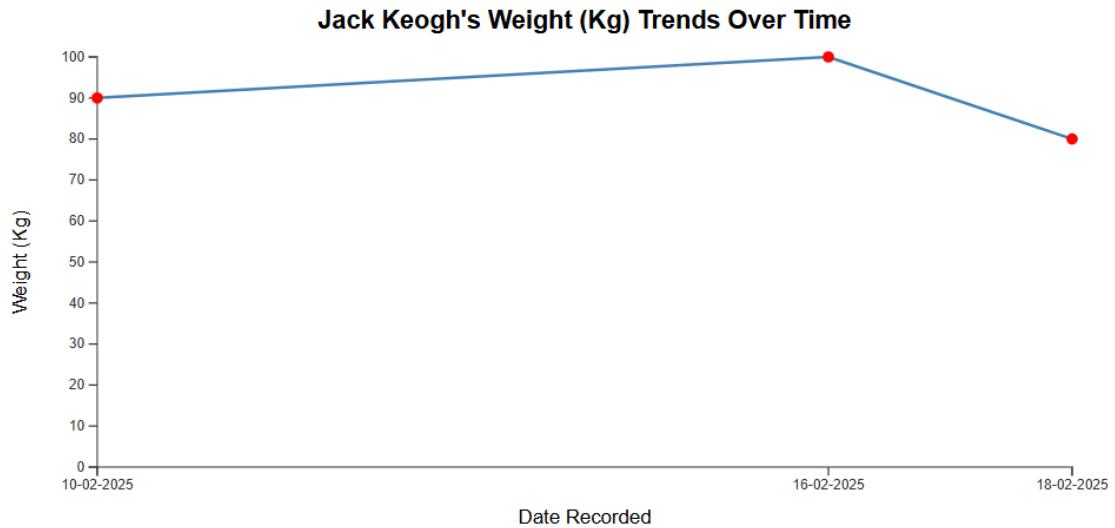


Figure 65: Line Chart displaying Jack Keogh's Weight Trends Over Time

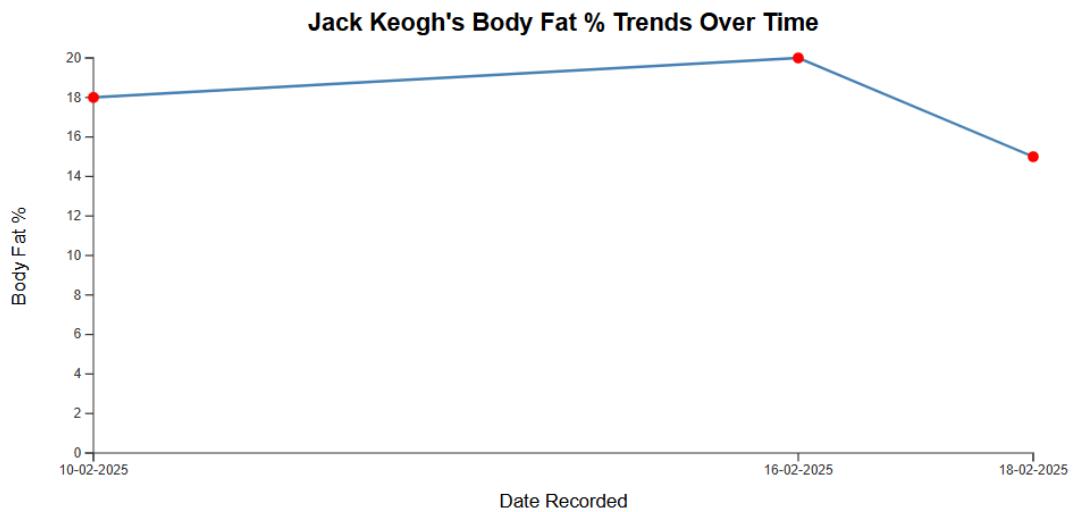


Figure 66: Line Chart Displaying Jack Keogh's Body Fat Percentage Trends Over Time.

6 Challenges Encountered in The Project

Throughout the research and development of this project, the author encountered a number of challenges in the project that he tried to overcome. Some of these challenges consisted of time management, implementing a suitable way of uploading team session statistics to the system and making all of the graphs more interactive on the club member statistics page.

Time Management: Time management was a crucial factor in the creation of this project as the author had a very busy schedule. The author had to complete assignments and study for exams in other modules as part of the requirements that the author needed to fulfil in order to obtain an honours degree in Software Development. The author is also currently employed and had to work shifts most weekends throughout the duration of this project. Due to these time-constraint factors, the author prioritised on completing the functionality of the project rather than implementing validation and stronger security in the web application.

Uploading Team Session Data to The System: The author implemented a form that allowed trainers to enter statistics that were recorded from a training session. The author demonstrated this functionality to his supervisor Brendan Watson. We discussed about a better approach of recording team session statistics to the system. The project was in its late stages which was quite challenging to implement a suitable approach to upload team sessions statistics in a short amount of time. The author researched into the Stat Sports Apex GPS Device that are worn by athletes during sessions and games to record metrics that can be used for analysis to improve the athlete's performance. The author discovered that the recorded metrics from each player can be exported to a CSV file. The author applied for demo of the GPS device from Stat Sports. Unfortunately, the author didn't get a reply from the company. As a proof of concept, the author implemented a feature which allowed trainers to upload a CSV file which contained sample metrics "Max BPM", "Resting BPM", "Average BPM" and "Calories Burned". The author realized that he should of took this problem into consideration in the earlier stages of the project as it would not of been feasible for a trainer to manually enter statistics recorded for every player that participated in a team session as a trainer could easily enter incorrect data due to human error and it would also be a very irritating task to carry out, especially if the trainers team consisted of a lot of players.

Graph Interactivity on The Club Member Statistics Page: The author implemented three graphs that displayed data insights on sports club members. The author wanted to update the values being displayed on all three graphs on the club member statistics page depending on the option selected by the club administrator. Unfortunately, the author wasn't able to update the values on all three charts and was only able to update the bar chart that displayed the number of males and female members recorded on the system.

7 Future Work

Uploading of team session data: The author implemented a feature that would allow the trainer to enter statistics on each player for a selected team session. The author researched into the Stat Sports Apex GPS discovered that it is possible that the recorded data can be exported to a CSV file. In future iterations of this project, the author can try to get possession of multiple Apex GPS devices and carry out a demo training session that consists of multiple participants wearing the devices to discover what type of metrics are recorded and also figure out how to export the data to a CSV file in order for the data to be recorded on the system. The author could also research more into different ways that data could be recorded from players and saved to the system.

Implementing Different Data Visualization Techniques: The author utilized bar graphs, line graphs and tables in his application to display data. In future iterations of this project, the author will try to apply other data visualizations techniques such as a choropleth map of Ireland which can display the number of members that come from each county.

Hosting The Application:

The web application must be deployed to a suitable hosting service for it to be available to all users. The author planned to host the web application through Azure cloud services but the author was worried about the costs of hosting the web application and the MySQL database. The author had prior experience of using Azure in which he hosted a database on a Microsoft SQL Server during his third-year group project. His team was provided with 100-euro worth of credit to host the database but after about six weeks, all of the credit was gone. The author needs to carry out more research into the resources that is required for hosting the web application and the MySQL database at a low cost.

Code Refactoring:

To correctly record the data that is gathered from the Apex GPS Device, The author needs to refactor the code for uploading team session statistics through a CSV file as the data from the Apex GPS Device will be different. He also needs to ensure that the data recorded is mapped to the correct players.

Improve Graph Interactivity on The Club Member Statistics Page:

On the club member statistics page of the application, the author wasn't able to update all of the values that was being displayed on the three graphs based on the option the club administrator selected. In future iterations of this project, The author will try to resolve this issue and enable club administrators to be able to view data insights on all three graphs that are implemented on the page based on the “All Members”, “Club Members”, “Players”, “Trainers” and “Admins” options the club administrator can select.

Improve User Experience:

To improve the user experience of the application. The author needs to implement search features in various sections of the application make it easier for users to find information on the sports club application.

The author also needs to improve the front-end design so that it is more user friendly for smaller devices such as smart phones and tablets.

Implementing Better Security and Form Validation:

The author mentioned that he utilised the Spring Security framework in which the author talked about the “filterChain” method in the “WebSecurityConfig” class which allowed the author to define who can make requests to certain controller endpoints based on their role. The author was more focused on completing the functionality requirements of the application and didn't define permissions for all of the controller endpoints. The author also didn't implement validation for any of the forms in the application. The author will need rectify this in future iterations of the project as these are necessary measures to prevent users from accessing confidential data and also prevent users from entering bad data which would cause the system to fail.

8 Conclusion

The aim of this project was to research into data analytics and modern technologies used in sports and their potential use for a sports club management application. The author then developed a sports club management web application that applied descriptive analytics and the potential use of a GPS device to record team session statistics. The application displays data on club members, club event attendance, team session attendance, players physical statistics and team session statistics using suitable data visualisation techniques to present data insights that could help identify trends or patterns that can influence informative decisions that could improve the sports club. In the late stages of the project the author researched into the Apex GPS device that are used by sports teams to record metrics of sessions and games which can be exported to a CSV file. Although the author didn't get the Apex GPS device, he believed that it can be utilised by sport clubs to record metrics on team sessions. The author then decided to implement a feature that can allow trainers to upload team session statistics from a CSV file to the sports club system as a proof of concept.

Before developing the application, the author carried out research into various sport applications used by sport clubs, the effect of modern technologies in sports, data analytics in sports, data visualisation techniques, how data analytics is utilised in software by developers and various web application frameworks and the features they provide. Carrying out this research educated the author more about data analytics and modern technologies used in sports and also suitable data visualization techniques that can be utilised in the sports club application. It also helped the author to define the applications requirements and also what web application framework the author should use to develop the web application.

The author was satisfied with using the Spring Boot framework as he was able to successfully implement most features of the application in a short amount of time which allowed the author to focus more on the implementation of the graphs to display the data using D3 and also add other features such as uploading team session statistics through a CSV file and also implementing a REST API that generates a PDF file containing the recorded statistics from a selected team session. Throughout the implementation process of the project, the author grew more competent in using Spring Boot and Thymeleaf. The

author also learned about new libraries such as D3, Apache Commons CSV, iText Core and Jackson.

Although the application needs improvement in several areas, the author is satisfied in what he was able to achieve within the amount of time. He does believe that it would have been better if he was able to get the Apex GPS device and investigate how metrics from sessions are exported to a CSV file. Even though the author didn't get the Apex GPS device, he believes that he was able to demonstrate the potential use of data analytics and modern technologies in a sports club management application by developing this application.

9 References

- Adam, D. (2022) ‘Science and the World Cup: how big data is transforming football’, *Nature*, 15 November, pp. 444–446. Available at: <https://doi.org/10.1038/D41586-022-03698-1>.
- Alex (2021) *Python Charts - Pie Charts with Labels in Matplotlib*. Available at: <https://www.pythoncharts.com/matplotlib/pie-chart-matplotlib/> (Accessed: 24 April 2025).
- Ali, M. (2024) *Top 26 Python Libraries for Data Science in 2025* | DataCamp. Available at: <https://www.datacamp.com/blog/top-python-libraries-for-data-science> (Accessed: 16 March 2025).
- Altair (no date) *Data Analytics and AI Platform* | Altair RapidMiner. Available at: <https://altair.com/altair-rapidminer> (Accessed: 16 March 2025).
- Ayooluwa, I. (2025) *A Complete Guide to Winston Logging in Node.js* | Better Stack Community. Available at: <https://betterstack.com/community/guides/logging/how-to-install-setup-and-use-winston-and-morgan-to-log-node-js-applications/> (Accessed: 3 February 2025).
- Bai, Z. and Bai, X. (2021) ‘Sports Big Data: Management, Analysis, Applications, and Challenges’, *Complexity*, 2021. Available at: <https://doi.org/10.1155/2021/6676297>.
- Bold BI (no date) *Embedded Analytics* | Seamless Integration for Your Business. Available at: <https://www.boldbi.com/embedded-analytics/> (Accessed: 24 April 2025).
- Briggs, B. (2019) *The Miami Heat are on a fast break to innovate* - Source. Available at: <https://news.microsoft.com/source/features/digital-transformation/miami-heat-fast-break-innovate/?msclkid=1a6b2d0081336434189b39ac807e65cb> (Accessed: 3 November 2024).
- Catapult (2024) *Sports Analytics: What is it & How it Improves Performance?* - Catapult. Available at: <https://www.catapult.com/blog/what-is-sports-analytics> (Accessed: 30 October 2024).
- ClubZap (no date) *ClubZap* | Club Communication And Payments. Available at: <https://clubzap.com/> (Accessed: 22 October 2024).
- Cogniteq (2023) *How Wearable Technology Is Changing Sports* | Cogniteq. Available at: <https://www.cogniteq.com/blog/how-wearable-technology-changing-sports-industry> (Accessed: 28 October 2024).
- Cote, C. (2021) *What Is Predictive Analytics? 5 Examples* | HBS Online. Available at: <https://online.hbs.edu/blog/post/predictive-analytics> (Accessed: 13 March 2025).

CueMath (no date) *Line Graph - Examples, Reading & Creation, Advantages & Disadvantages*. Available at: <https://www.cuemath.com/data/line-graphs/> (Accessed: 24 April 2025).

D3 (no date) *What is D3? | D3 by Observable*. Available at: <https://d3js.org/what-is-d3> (Accessed: 26 April 2025).

Data Overload (2023) *Challenges and Limitations of Sports Analytics: What We Still Don't Know | by Data Overload | Medium*. Available at: <https://medium.com/@data-overload/challenges-and-limitations-of-sports-analytics-what-we-still-dont-know-d6e1d34a445a> (Accessed: 2 November 2024).

Databrain (2023) *Unlocking the Power of Data: 9 Key Benefits of Embedded Analytics*. Available at: <https://www.usedatabrain.com/embedded-analytics/benefits> (Accessed: 12 March 2025).

DataTeer (no date) *What is Embedded Analytics? Benefits, Tools/Software & Examples - DataTeer*. Available at: <https://www.datateer.com/blog/what-is-embedded-analytics/> (Accessed: 12 March 2025).

Everite (2024) ‘AI-Powered Predictive Analytics: The Key To Smarter Software Projects - Everite Solutions’, <https://everitesolutions.com/> [Preprint]. Available at: <https://everitesolutions.com/ai-powered-predictive-analytics-the-key-to-smarter-software-projects/> (Accessed: 14 March 2025).

GeeksforGeeks (2024a) *Top 10 Frameworks for Web Applications in 2025 - GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/top-frameworks-for-web-applications/> (Accessed: 3 November 2024).

GeeksforGeeks (2024b) *What is Angular ? - GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/what-is-angular/> (Accessed: 4 November 2024).

GeeksForGeeks (2024) *What is Python? Its Uses and Applications - GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/what-is-python/#uses-and-applications-of-python> (Accessed: 16 March 2025).

GeeksforGeeks (2025) *Angular Tutorial - GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/angular-tutorial/> (Accessed: 24 April 2025).

Goldstein, P. (2017) *Billy Beane: Data Analytics Has Transformed Baseball | BizTech Magazine*. Available at: <https://biztechmagazine.com/article/2017/09/splunk-conf2017-oakland-billy-beane-says-data-analytics-has-transformed-baseball> (Accessed: 1 November 2024).

Hassan, M. (2024) *Predictive Analytics - Techniques, Tools and Examples*. Available at: https://researchmethod.net/predictive-analytics/#google_vignette (Accessed: 16 March 2025).

Hudl (no date) *About our Company • Hudl*. Available at: <https://www.hudl.com/about> (Accessed: 24 October 2024).

IBM (2021) *What Is Data Visualization? | IBM*. Available at: <https://www.ibm.com/topics/data-visualization> (Accessed: 2 November 2024).

IBM (2022) *What is Predictive Analytics? | IBM*. Available at: <https://www.ibm.com/think/topics/predictive-analytics> (Accessed: 13 March 2025).

International Olympic Committee (2024) *AI and tech innovations at Paris 2024: A game changer in sport*. Available at: <https://olympics.com/ioc/news/ai-and-tech-innovations-at-paris-2024-a-game-changer-in-sport> (Accessed: 29 October 2024).

Jovanovic, M. (2024) *Game-Changing Sports Data Visualization Examples*. Available at: <https://wpdatatables.com/sports-data-visualization/> (Accessed: 3 November 2024).

Khorasani, V. (2024) *Gym Members Exercise Dataset*. [dataset] Kaggle. Available at: <https://www.kaggle.com/datasets/valakhorasani/gym-members-exercise-dataset/data> (Accessed: 26 April 2025).

King, T. (2025) ‘The 20 Best Embedded Analytics Tools and BI Software for 2025’, 1 January. Available at: <https://solutionsreview.com/business-intelligence/the-best-embedded-analytics-tools-and-bi-software/> (Accessed: 13 March 2025).

Kumar, A. (2024) *Web Development Frameworks: A Guide | Built In*. Available at: <https://builtin.com/articles/web-development-frameworks> (Accessed: 3 November 2024).

Lin, P. (2023) *10 Types of Data Visualization You Can Use for Your Reports*. Available at: <https://www.computer.org/publications/tech-news/build-your-career/types-of-data-visualization-for-reports> (Accessed: 2 November 2024).

Log4j (no date) *Apache Log4j :: Apache Log4j*. Available at: <https://logging.apache.org/log4j/2.x/index.html> (Accessed: 3 February 2025).

Mazza, L. (2018) *The Current and Future Impact of Virtual Reality on Sports - The Hub*. Available at: <https://hub.fullsail.edu/articles/the-current-and-future-impact-of-virtual-reality-on-sports/> (Accessed: 29 October 2024).

MDN Web Docs (2025) *Django introduction - Learn web development | MDN*. Available at: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction> (Accessed: 3 November 2024).

MDN Web Docs (no date) *Ajax - MDN Web Docs Glossary: Definitions of Web-related terms | MDN*. Available at: <https://developer.mozilla.org/en-US/docs/Glossary/AJAX> (Accessed: 26 April 2025).

Mostert, B. (2023) *Embedded Analytics Defined*. Available at: <https://www.oracle.com/business-analytics/embedded-analytics/> (Accessed: 12 March 2025).

- Practice (2023) *10 Sports Coaching Apps for the Win* | Practice. Available at: <https://practice.do/blog/sports-coaching-apps> (Accessed: 23 October 2024).
- Pykes, K. (2022) *Sports Analytics: How Different Sports Use Data Analytics* | DataCamp. Available at: <https://www.datacamp.com/blog/sports-analytics-how-different-sports-use-data-analysis> (Accessed: 22 October 2024).
- Ricky, A. (2019) *How Data Analysis In Sports Is Changing The Game*. Available at: <https://www.forbes.com/councils/forbestechcouncil/2019/01/31/how-data-analysis-in-sports-is-changing-the-game/> (Accessed: 25 April 2025).
- Scutaru, C. (no date) *Alternatives to Pie Charts, with FREE Cheat Sheet Infographic*. Available at: <https://data-xtractor.com/blog/data-visualization/alternatives-to-pie-charts-2/> (Accessed: 26 April 2025).
- Springer, S. (2016) *How The Patriots Use Data To Try To Improve The Fan Experience* | Morning Edition. Available at: <https://www.wbur.org/morningedition/2016/12/22/kraft-analytics-group> (Accessed: 1 November 2024).
- Stat Sports (no date a) *Clients - STATSports*. Available at: <https://pro.statsports.com/statsports-clients/> (Accessed: 26 April 2025).
- Stat Sports (no date b) *STATSports Sonra Lite - STATSports*. Available at: <https://pro.statsports.com/sonra-lite/> (Accessed: 23 April 2025).
- Stevens, E. (2023) *What Is Data Visualization? A Complete Introductory Guide*. Available at: <https://careerfoundry.com/en/blog/data-analytics/what-is-data-visualization/> (Accessed: 2 November 2024).
- Tableau (no date) *Embedded Analytics | Create seamless analytics experiences for your customers*. Available at: <https://www.tableau.com/products/embedded-analytics#key-features> (Accessed: 13 March 2025).
- Team EMB (2024) *Technology in Sports: Enhancing Performance and Experience*. Available at: <https://blog.emb.global/technology-in-sports/> (Accessed: 28 October 2024).
- Teamer (no date) *Teamer*. Available at: <https://www.teamer.net/> (Accessed: 23 October 2024).
- The University of Melbourne, S.C.C. (no date) *Why you shouldn't use pie charts*. Available at: https://scc.ms.unimelb.edu.au/resources/data-visualisation-and-exploration/no_pie-charts (Accessed: 18 March 2025).
- Thrane, C. (2023) *Sports data, regression analysis, and interaction effects* | by Christer Thrane | Medium. Available at: <https://medium.com/@christerthrane/sports-data-regression-analysis-and-interaction-effects-4894e40a5509> (Accessed: 13 January 2025).

- W3schools (no date) *Introduction to Django*. Available at:
https://www.w3schools.com/django/django_intro.php (Accessed: 3 November 2024).
- Wright, R. (2023) *Why Is Logging An Important Part Of Software Development? - Data Loggers*. Available at: <https://dataloggersdefined.com/why-is-logging-an-important-part-of-software-development/> (Accessed: 3 February 2025).
- Zeichick, A. (2024) *What Is Spring Boot?* Available at:
<https://www.oracle.com/database/spring-boot/> (Accessed: 3 November 2024).