

Research Document



SportsPA

Name: Gerard Dobbs

Student Number: C00196843

Course: Bachelor of Science (Hons) Software Development

Supervisor: Mr. Paul Barry

Date: 18/04/2018

Abstract

This document has been produced to detail the research which has been carried out while completing my project.

The document will detail the purpose of the application and other related systems that are in existence.

It will document the different technologies which have been researched and why the chosen technologies are suitable for this project.

Table of Contents

Abstract	1
Table of Contents	2
1.Introduction	3
2. Background	4
3. Related Systems for Sports Management App	5
3.1 Performa Sports	5
3.2 Statapult	9
3.3 Dartfish EasyTag	14
3.4 GAA Scores Stats Plus	16
3.5 Conclusion	19
4. Technologies - Backend for Sports Management App	20
4.1 Firebase	20
5. Technologies - Front End for Sports Management App	21
5.2 Dart	23
5.3 Ionic	24
5.4 PhoneGap	24
6. Related Systems for SportsPA	25
6.1 Hudl	25
6.1.1 Sportscodel	25
7. Python libraries for parsing XML	26
7.1 xml.etree.ElementTree	26
7.2 xml.dom.minidom	28
7.3 Conclusion	30
8. Python Visualization	31
8.1 Bokeh	31
8.1 Matplotlib	32
9. Conclusion	33
10. References	34

1.Introduction

The contents of this research document will outline all the research that was conducted in relation to this 4th Year Software Development project.

It will start off by giving a background to the project. This will outline why I decided to do the Sports Management Application. It will detail how this app will be of benefit to sports teams. The unique selling points the app will possess will also be explained.

Related systems that are available will then be outlined. An explanation will be given, supported by screenshots, of how each system works. It will outline the benefits of each specific system. I will then explain any items that are not included and which i propose to include in my application. The cost of each system and what devices they are available on will be explained.

While conducting this research there was a genuine need for an application closely linked, but different in many ways, to the initial goal of the project. This was identified by Johnny Bradley, a lecturer at IT Carlow who is involved in Sports Performance Analysis. Currently students, studying Performance Analysis, are using Software which produces output in the format of XML files. This format is not easily readable and very difficult to analyse. There is software available, at a very high price, which produces output as a dashboard, which is easily interpreted. Due to this real need the focus of the project changed, with a view to satisfying the needs of Johnny Bradley and his students.

Therefore the project will be led by requirements specified by Johnny Bradley. These requirements may change as the project progresses and for this reason small iterations will be provided. The title of the project has also been altered to reflect the change in requirements.

Chapters 3 to 5, inclusive will detail research which was carried out for the original project specification. Although this research is mostly irrelevant with regards to the new project, it has given me knowledge which can be taken forward as I further my career. It has also given me an insight into how to conduct relevant research. Chapters 6 to 9 will detail research carried out which is relevant to the new project, SportsPA.

The first task in the project will be to parse the XML files into a format that will be more useful for implementing visualizations. Chapter 7 will detail the ElementTree and Document Object Model (DOM) API's. The conclusion to this chapter will explain why the chosen package was selected.

2. Background

Having been an avid lover of playing sports throughout my life, I then turned to watching sports when my playing career finished. The sports I have played the most, and subsequently view now on a regular basis, are GAA and Rugby. While attending a lot of matches I've seen supporters trying to record scores on paper or else getting the scores from other supporters. This led me to develop my first Android App in the summer of 2015 to keep both the time and scores at GAA or Rugby matches. I published this simple App on the Playstore.

Statistics have become a very important part of sporting life. Originally something used by bigger sports club and teams, it is now something that all teams try to use. They help team managers and coaches get a clear picture of trends in matches. This then allows them to improve on any events that lead to poor performance and results. It also allows the team to continue implementing elements of their play that are getting positive results.

While attending matches in both GAA and Rugby I have noticed a growing trend in the recording of statistics. A lot of the teams I have seen recording events have done so by using pen and paper. In order to view statistics over a period of time the sports management team need to trawl through data recorded on paper to seek out trends. The use of technology would greatly enhance both the recording of events and the analysing of statistics.

This project aims to provide that needed technology and make it accessible and usable to teams of all levels. This project commenced by speaking with people involved at a management level with GAA and rugby teams. The consensus was that there is a fast growing trend towards using statistics. The events recorded by teams varies. I also spoke with two people involved in sports statistics. The first was involved in recording rugby statistics for Tullow R.F.C and Naas R.F.C. These recordings were achieved by videoing matches and later on going through the video and recording the events. The second person I spoke to is teaching a masters in Sports Performance at IT Carlow.

Following these conversations, a gap in the market was identified. This was the ability for a coach or assistant to record events on a device and for the manager to be available to view the statistics in real time on another device. This would give a sports management team the ability to make quick and important decisions.

Another observation I have made while attending matches, and as a user of social media, was facebook page administrators writing posts during games to update remote supporters on events. I plan to incorporate into the App, the ability to automatically send notifications of certain events as they are recorded. These events will include scores, yellow cards, etc..

Having really enjoyed working on my first Android App, coupled with my love of sports and the gaps I see in the market, this project was an easy choice for me.

I commenced my research by speaking with people involved in coaching, video analysis and Sports Performance analysis. As part of this research, I spoke with Johnny Bradley. Johnny is a lecturer in Sports Performance analysis at IT Carlow. Johnny was very receptive to the idea I presented to him and was very eager to provide any assistance he could. I continued to meet Johnny and at one of these meetings Johnny came up with an idea where he believed there was a gap in the market. This idea entailed providing real time analysis. This would involve a coach recording KPI's on a device while a match was taking place. At the same time another coach could be watching the data being updated in real time on a different device. This would enable quick and informed decisions to be made.

This idea sounded very exciting and after consultation with Paul Barry it was decided to take this new approach.

3. Related Systems for Sports Management App

This Chapter will outline the Research I have carried out on systems available which can be used for the purpose of recording statistics.

3.1 Performa Sports

Following my conversations with those involved in Sports statistics the System that was mentioned most was Performa Sports.

Performa Sports allows the user create their own unlimited number of Key Performance Indicators.

“Key Performance Indicators, also known as KPI or Key Success Indicators, help managers and employees gauge the effectiveness of various functions and processes important to achieving organizational goals[1].”

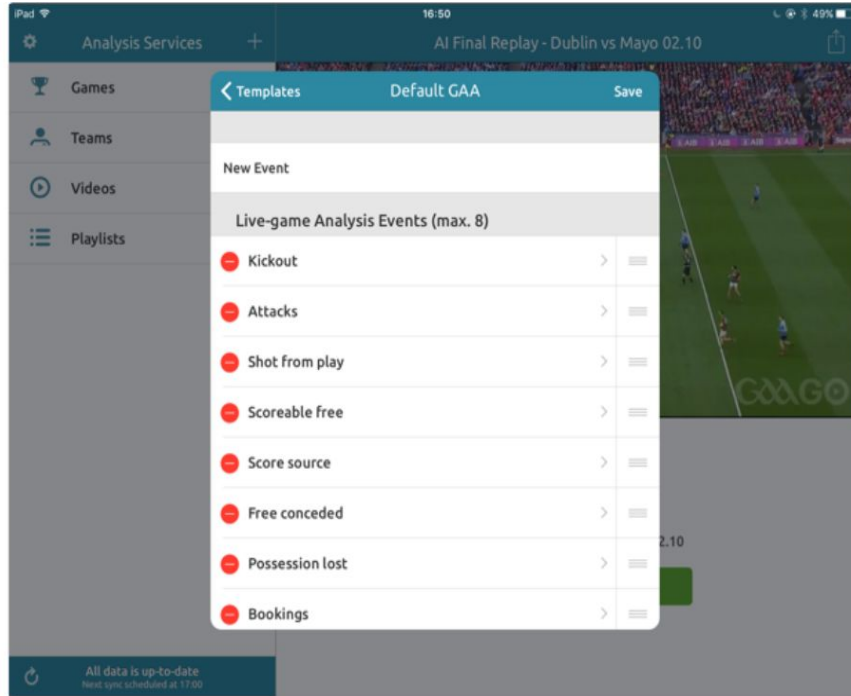


Figure 3.1.1 - Create your own KPI's.

Squad Profiles can be created by selecting players, giving them squad numbers and uploading a photo of each player. Player profiles can also be created for opposition players. Before the start of each game, players are selected from the already created squad and given their position on the team. As substitutions are made, players can be replaced by other squad members.

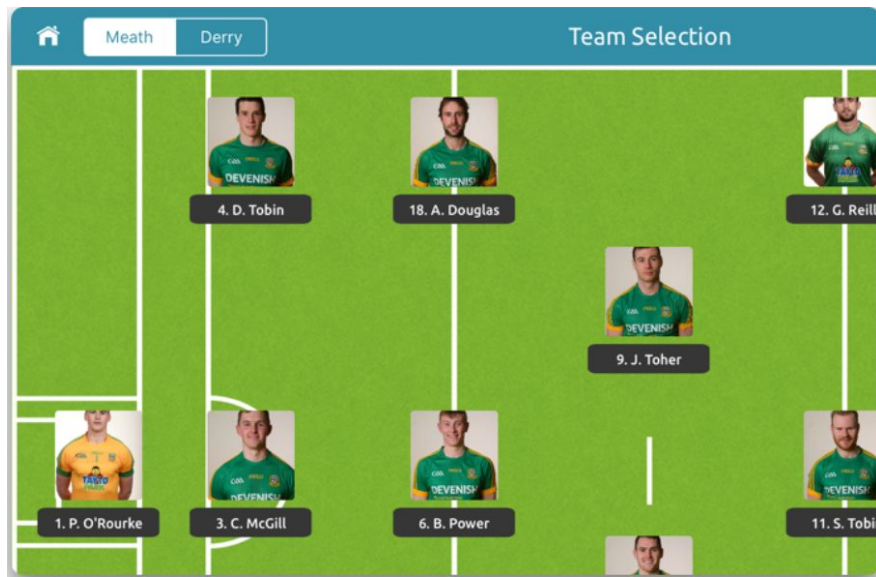


Figure 3.1.2 - Select Team from Squad.

When the game starts the player numbers appear on the left of the screen beside the KPIs that have been created. As events occur the player and KPI are selected. The events automatically appear on the screen along with the associated time. The number of occurrences of each KPI also appear beside the KPI itself. Having this data at hand and having the ability to filter results means it's easier to make key tactical decisions immediately.

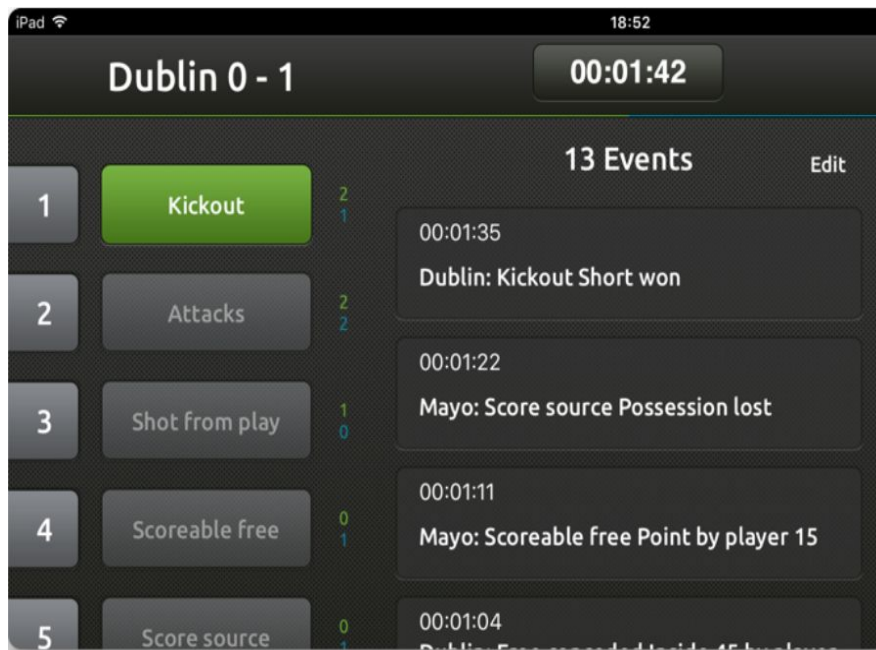


Figure 3.1.3 - Select Player Number & KPI.

This system also has the ability to video games which can later be used to tag events and record statistics. The video can also be synced with any data that has been recorded in order to view the KPI's recorded while watching the video. It is also possible to import videos and tag events to record statistics. A recorded KPI can be selected and the video will proceed to that point.

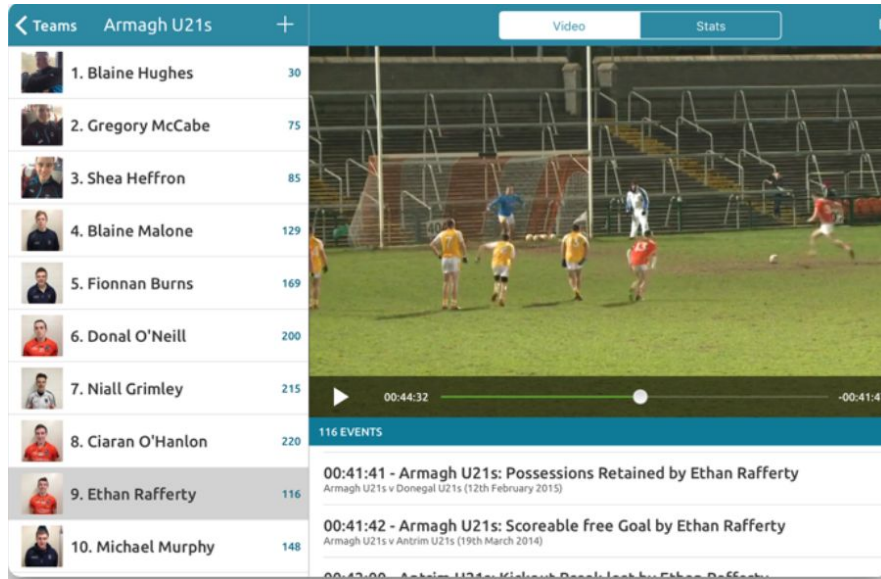


Figure 3.1.4 - KPI event played out on video.

The Performa Sports system also allows the user to use drawing tool, while relaying the video, in order to make it easier to get a point through to players.

“Performa Sports Online Analytics Platform integrates seamlessly with the iPad application allowing you to sync game data to your secure Cloud based account[2].”

The Cloud Analytics Platform makes it easier to share video and statistics with interested parties.

This system has been built for use on an iPad. The cost is on a yearly rate with the basic package costing €399 per year. The next, and most popular package, costs €799 per year. The final package can be got for €1799 per year. This ties the user into a contract, which will not be a feature of The Sports Team Management App. Therefore the App in this project will be more accessible for all sports teams.

This system is very detailed and relatively easy to follow for the basic package. As packages get more expensive, and therefore more detailed, The interface gets a little more complicated.

The System does not have the ability to allow another user, such as the manager of the team, view the statistics in real time on their own device. This feature is what the Sports Team Management App will use as its unique selling point.

3.2 Statapult

Statapult was found to be the application which was closest in content to the Sports Team management App, which is the subject of this project. Statapult is an iOS application, available for use on an iPad. It performs analysis on statistics from GAA, Rugby and Soccer games. There is a separate application for each sport and it performs analysis on selected key performance indicators (KPI's). There are 10 preset KPI's along with the option of selecting another 10 KPI's, from 20, which can be configured as per requirements.

The KPI's which can be selected by the user are location based. These KPI's record the location on the pitch where the event occurred. The pitch is broken down into zones for this analysis.

When the application is opened the user is taken to the game setup screen (Fig1). Here the user has the choice of entering a new team squad or selecting from a previous squad. The match details may also be entered along with the name of the opposing team.

Figure 3.2.1 - Enter Players names or select from previous squad

This game data is then saved. The user can then go to the system settings where the KPI's to be recorded can be selected (Fig 3.2).

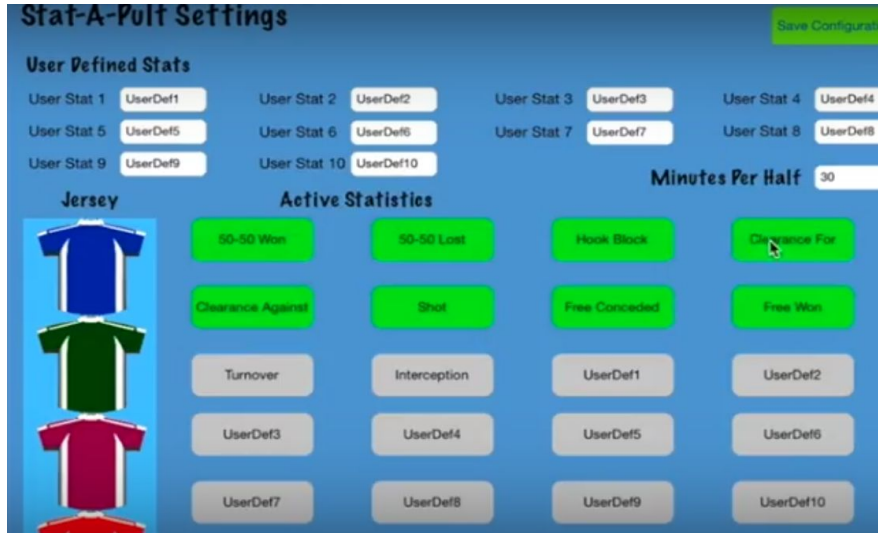


Figure 3.2.2 - Select KPI's to use in game.

When the configuration is saved the game timer can be started and the game is then active.



Figure 3.2.3 - Select player linked to KPI event.

An event then occurs involving a player from your team. That player is then selected which will result in the associated KPI's appearing on screen.



Figure 3.2.4 - Select KPI to link to selected event.

The KPI is selected and will be recorded. A stat overview screen may then be selected. This screen will display all currently recorded events.

The screenshot shows a table with 8 columns: Player Name, Number, 50-50 Won, 50-50 Lost, Hook Block, Shot, Free Conceded, and Free Won. The table lists 11 players. Player 9 is highlighted in green, indicating a recorded event. At the bottom, there are controls for 'Order Stats By' (set to 'Number'), 'End Game', 'Reset Stats', and 'Export Current Game Data'.

Player Name	Number	50-50 Won	50-50 Lost	Hook Block	Shot	Free Conceded	Free Won
	3	0	0	0	0	0	0
	4	0	0	0	0	0	0
	5	0	0	0	0	0	0
	6	0	0	0	0	0	0
	7	0	0	0	0	0	0
	8	0	1	0	0	0	0
	9	1	0	0	0	0	0
	10	0	0	0	0	0	0
	11	0	0	0	0	0	0
		1	1	0	0	0	0

Figure 3.2.5 - Recorded KPIs.

These stats can then be viewed by whatever order the user desires. To record a score or wide the user selects an area on the pitch. This will bring up the KPIs relevant to a score or wide.



Figure 3.2.6 - Record Shot/Wide on selected area of the pitch

When the relevant KPI is selected, the player involved in that event is then selected. This event will then be added and can be viewed on the score summary screen. In order to record an event by the opposition, there is the option of adding 2 customizable KPI's. This will allow the user to analyse 2 opposition events such as scores. These can be recorded as already described.

The shot success rate for each team can then be viewed. This will display the success rate in each zone. This is displayed in the format 1/1, which indicates 1 successful shot from 1 attempted shot.

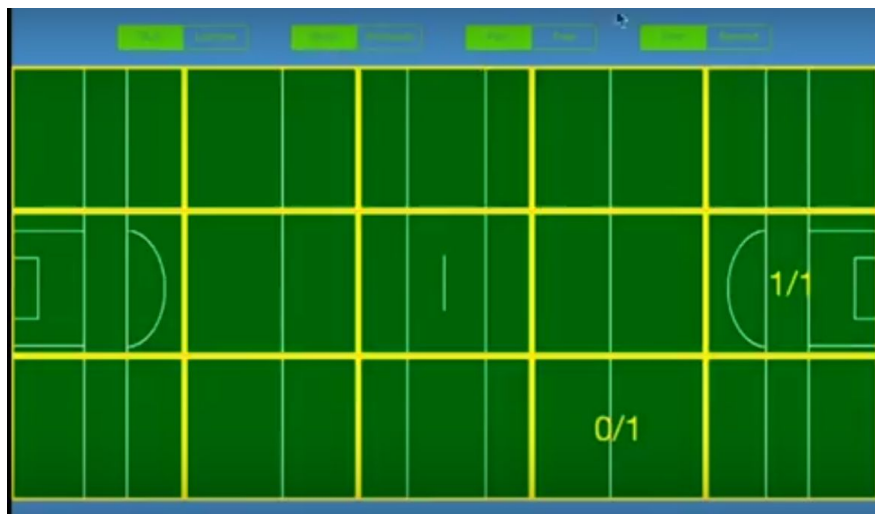


Figure 3.2.7 - Display of shot success rate for each area of the pitch.

Kickouts won and lost can also be recorded and viewed in this manner.

When the match is over, the resulting statistics can be emailed to interested parties,

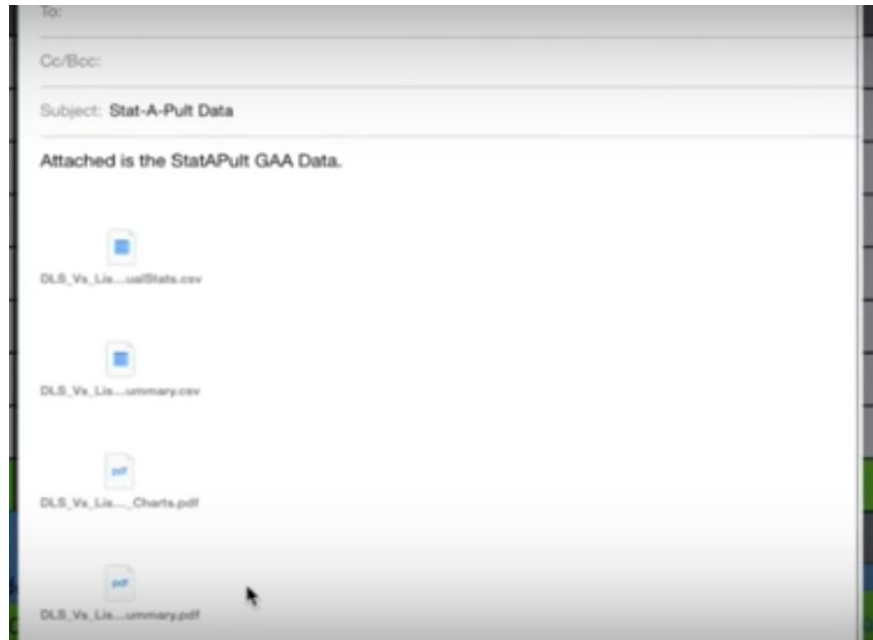


Figure 3.2.8 - Export results via email.

This application is available from the app store for \$14.99. It is only available for the iPad. There are many details in the app which match closely The Sports Team Management App.

However, the statistics stored are not persistent. It is proposed that my application will use a database to store all statistics. These can then be analysed to give patterns of play. As the database grows, so too will the accuracy of the analysis.

The System does not have the ability to allow another user, such as the manager of the team, view the statistics in real time on their own device. This will be the main Unique Selling point of the Sports Team Management App.

The Sports Team Management App will also automatically push notifications of selected events. For example, when a score occurs, the details of the score and scorer will be sent to relevant Facebook and Twitter accounts.

3.3 Dartfish EasyTag

The Dartfish EasyTag can be used to record statistics for any sport. The application includes a customizable panel which must be set first. This panel can include from 9 to 39 buttons. Each button will represent a KPI.



Fig 3.3.1 - Create Panel with preferred KPI's.

This example displays a panel for a soccer match. The user has set up an 8x2 panel which records statistics for the match. As an event occurs, the corresponding button is selected and the count of the occurrences of that KPI is recorded. The panel, in the example above, has events in blue for one team and in red for the other. A match is then set up by entering the names of the teams. This match can then be associated with the newly constructed panel or with a previous panel. When the game clock is started the buttons can be selected to make recordings.

10 events	
00'18	
stop	
Fouls (1)	Foul (2)
Shot On T (0)	Shot on T (1)
Shots Off T (1)	Shot off T (0)
Corners Good (1)	Corner Good (0)
Corners Poor (0)	Corner Poor (1)
Entry Att 1/3 (1)	Entry Att 1/3 (1)
Throw In Won (1)	Throw in Won (0)
Throw In Lost (0)	Throw In Lost (0)

Fig 3.3.2 - Number of each KPI recorded is displayed.

The statistics are displayed, in brackets on the relevant button. When the match is over the recorded KPIs can be viewed on a review screen. There is an option to view the statistics by category or by time. The results can also be exported by email for viewing through another medium or to other interested parties.

This application is available from the app store for €4.99. It is only available for the iPad and iPhone. This is a very basic app. Although it is customizable with regard to the KPIs recorded, there is no way to connect a player to particular KPIs. The statistics can be emailed but are not persistent. There is no way to build up a database of results for further analysis. There is also no way to send push notifications of events as they occur. These are all elements that will be included in the Sports Team Management App.

3.4 GAA Scores Stats Plus

This is an android app which can be downloaded from the playstore. It is intended for use on android phones.

This app allows the user to record predetermined KPIs for GAA games. It does not allow the user to determine what KPIs to use. There are separate tabs where the players on each team can be recorded. These tabs also allow the substitution of players during game time.



Fig 3.4.1 - Players selected for the team.

The match tab allows the user to set a clock for the match duration and to record a goal or point for either team. It also allows the option of entering a stats screen. There is a different stats screen for each team. Statistics recorded to date will also appear. There is an option to share match events, as they occur, via social media.

DUBLIN		KERRY	
1-16 (19)		0-20 (20)	
	Total	from play	frees 45/65 pens
Goals	1	1	0
Points	16	12	4
Wides	9	8	1
Out for 45/65	0	0	0
Off Posts	1	1	0
Saved	1	1	0
Saved & out for 45/65	0	0	0
Short	3	1	2
TOTAL SHOTS:31 Scored:17 (54%)			
Shots from play:24 Scored:13 (54%)			
Shots from free/pen/45/65:7 Scored:4 (57%)			
Frees conceded	17		
Turnovers Won	11	36%	
Turnovers Lost	19	63%	
Total Own Puck/Kick Outs***	28		
Won	24	88%	
Lost	3	11%	

Figure 3.4.2 - All recorded KPI's displayed on the screen.

The stats screen gives an outline of all the events that have been recorded. It displays how many times each event has occurred. The details can be shared via social media.

Ballybuggles scorers 3-14(23)	
player	Total goals points placed balls goals points wides/masses total placed
Joshua	1 2 0 0 2 0
Jimmy	0 4 0 3 1 0
Hugh	1 1 0 0 1 0
Jack	0 3 0 0 3 0
Colm	1 0 0 0 0 0
David	0 2 0 0 2 0
Neil	0 1 0 0 4 4
Alan	0 1 0 0 0 0
Kevin	0 0 0 0 1 0

Killikely scorers 0-15(15)	
player	Total goals points placed balls goals points wides/masses total placed
--	0 15 0 9 9 5

Fig 3.4.3 - Scorers Statistics displayed.

The scorers tab keeps a record of goals, points and misses which have been recorded for each player. Again these details can be shared.

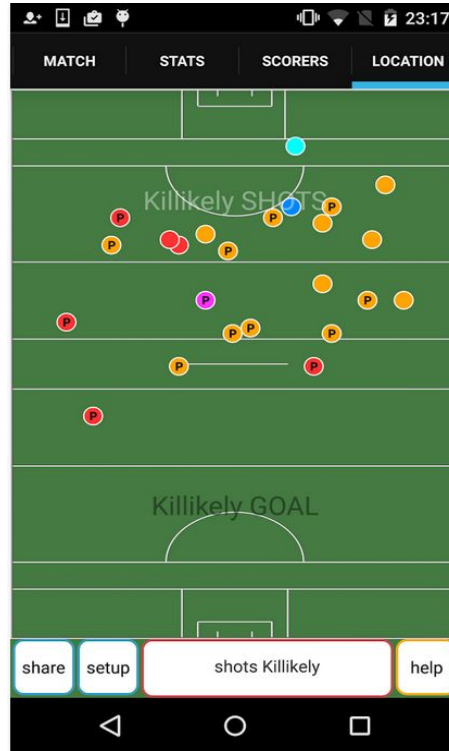


Fig 3.3.4 - The Location of where each event occurred.

There is also a tab to display graphics of the position on the pitch where events occurred. The opportunity to share this screen is also available.

This app is available for free on playstore. It provides a lot of the functionality that The Sports Team Management App will have. However, like the other apps researched it does not store the results once a new match is started. Also the screen look quite cluttered. To recorded some events the user must go through 3 different interfaces, which can be very tedious and confusing.

3.5 Conclusion

There are varied range of software packages and applications to record statistics for sports teams. They all differ in what they offer the client, what devices they target and the price range. Some of the applications store statistics whereas with others the details are lost when the application is closed. A few applications forward the statistics via email at the end of the game, while one can forward the email while the recording of the data is in progress.

There are two unique selling points that the sports management app will have compared to those researched. The main one is the ability for a manager to view statistics on his device while his assistant is still recording. From the research carried out this facility is not on offer. The other usp is the ability to automatically update social media with selected events as the occur. Some of the researched systems allow the user to update social media manually.

4. Technologies - Backend for Sports Management App

4.1 Firebase

“Everything you need to build and grow your app, all in one place[9].”

Firebase is a Mobile Backend As A Service and eliminates the need to manage your own backend. It's an integrated suite of tools and supports the development of Apps in Android, iOS and the Web. The set up and management of the app is done through a central dashboard.

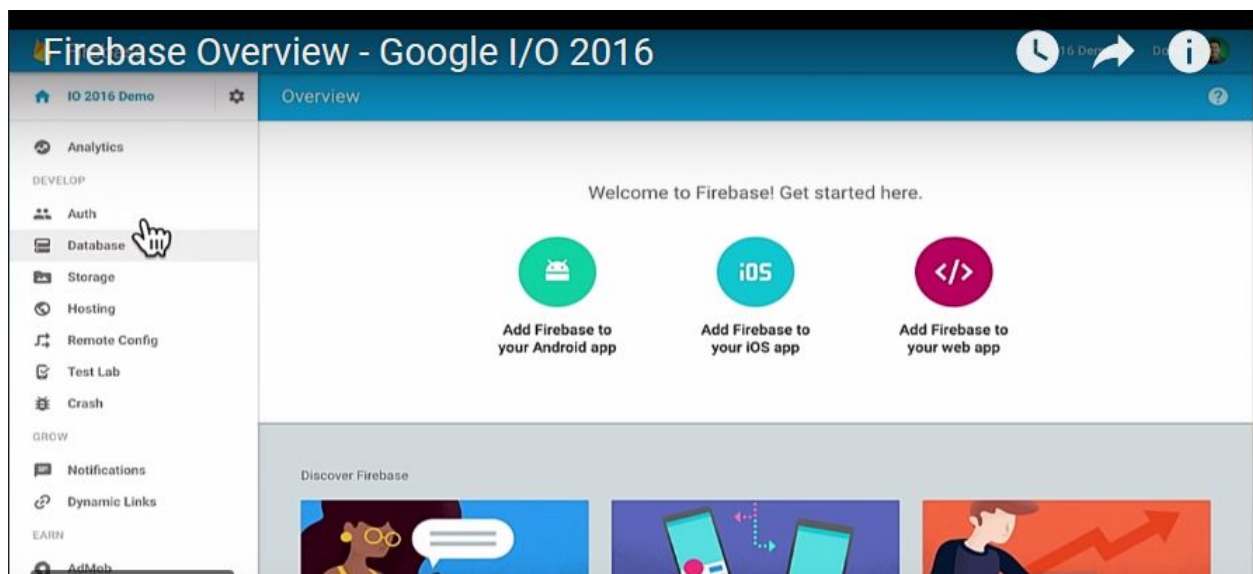


Figure 4.1.1 - Firebase Dashboard[10]

It provides a backend server for data and file storage. Data can be stored and synced to users in real time with no need to send requests to access the data or to refresh. It uses a NoSQL database with data stored as JSON Objects. This makes collaboration between users very easy and seamless and can be achieved on any device. If a user goes offline while sending data, it is stored in memory and then, when back online, the data is automatically synced. This allows the app to remain responsive.

Firebase hosting allows you to host HTML, CSS, JavaScript and images for web applications.

It also scales automatically ensuring that as the users and data grow then Firebase will scale up to accommodate the increase.

Authentication is also provided by Firebase and can be carried out via email, phone number, social media or any existing authentication systems.

Firebase also contains an analytics tool, based on Google Analytics but designed specifically for use on mobile Apps.

5. Technologies - Front End for Sports Management App

Originally mobile apps were developed using native Software Development Kits (SDKs) for iOS and Android. These used Objective-C and Java Languages respectively. This meant that when creating app, in order to get reach as large an audience as possible, separate apps had to be developed for each platform. Access to services such as the camera is provided, on both platforms by the app connecting with each platform. Widgets are used to render the interface to the screen and events are recorded and passed back to the widget.

Mobile app development then moved to cross platform development. This meant coding just once, using HTML, CSS and Javascript, and being able to deploy to both iOS and Android. This can lead to performance issues as there is a need for a bridge in order to allow Javascript code to communicate with the native services. For many apps this may not be an issue and would be greatly outweighed by the ability to have a single code base. The main cross platform technologies are Flutter, Phonegap and Ionic.

5.1 Flutter

Flutter is a Google mobile app Software Development Kit (SDK) for developing high performance, high fidelity apps[5]. It always the development of apps, using a single codebase, for both Android and iOS. Flutter follows a reactive methodology when building user interfaces.

“Reactive programming is programming with asynchronous data streams[6].”

Flutter allows you to build your User Interface (UI) using widgets, similar to the way UI's are built in native apps. Flutter use their own set of widgets that are customizable. Widgets give a good user experience by providing a good consistent look and feel, irrespective of the screen size. The widgets will change the view depending on the current state and likewise will be rebuilt as its state changes. However it is only the changes that are rebuilt, making the change much faster.

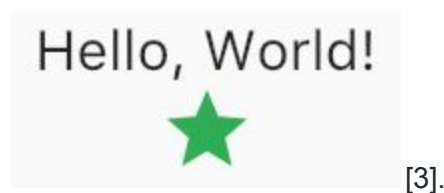
The use of widgets in Flutter is similar to developing in a native environment, by employing Material Design for Android and Cupertino for iOS. It also implements platform specific events and tools. However there are a number of differences in using widgets in native apps and using them in Flutter. Each widget in Flutter specifies its own layout model as opposed to native apps having their own set of layout rules[3].

There are both stateless and stateful widgets in Flutter. Stateless widgets are static and cannot be changed. On the other hand stateful widgets hold their state and can be changed seamlessly.

The following code, demonstrates how practically everything in Flutter is a widget.

```
new Center(  
  child: new Column(  
    children:[  
      new Text("Hello World!")  
      new Icons(Icons.star, color: Colours:green)  
    ]  
  )  
)[3].
```

The result of running this code is as follows:



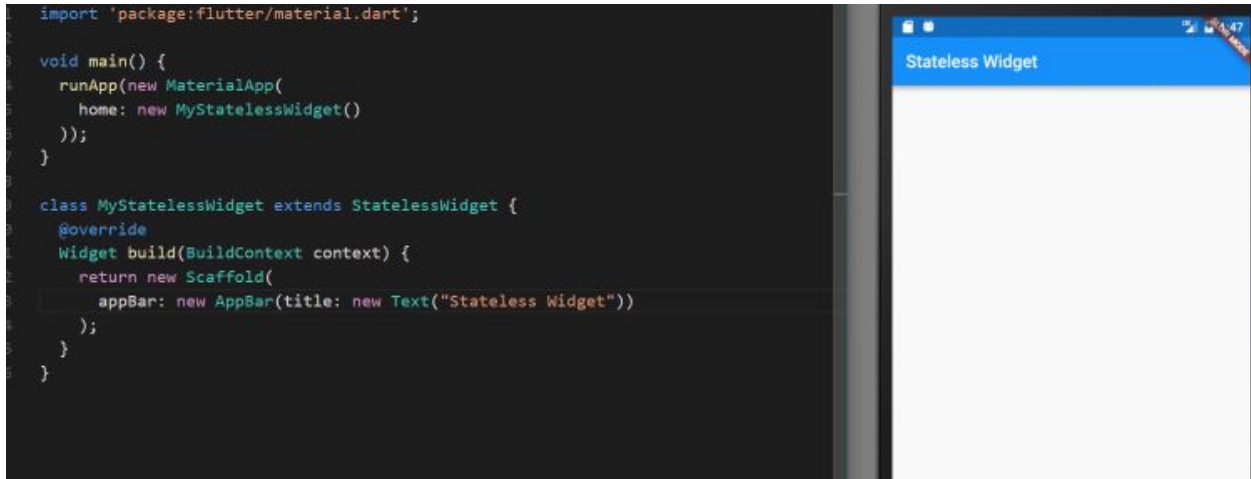


Figure 5.1.2[7] - Widgets

The widgets can be built in layers over each other as the app constructs a tree like structure. For example, the code in Figure 5.1.2, will use a text widget built on an app bar widget. The app bar widget is built on a scaffold widget.

An interesting feature of Flutter is Hot reloads, allow the user to check the impact of any code changes *immediately*, without the need for a restart. As the app is running, a Hot Reload will load any changes immediately and carry on running from where it left off. This feature allows the developer to view the app and test it as code is written.

Flutter can also be integrated with Firebase. This makes it very easy to build beautiful apps which are serverless.

5.2 Dart

Dart is a google programming language. It was originally developed within google as a direct competitor with Javascript for use on the web. Google then changed its focus and instead of competing with JavaScript, Dart code was compiled into JavaScript.[8]. It was then mainly used within Google by the AdSense and AdWords teams in developing their web apps. One of its main uses today is in developing Flutter Apps. When used to develop apps it is compiled, Ahead of Time (AOT) into machine code.

5.3 Ionic

Ionic is used to build cross-platform hybrid mobile applications. It is an open source framework, built on angular, allowing the developer to use web technologies such as HTML5, CSS and JavaScript to build applications for Android, iOS and the Web. Ionic concentrates mainly on the UI of the application, and because it's built on Angular, allows for the creation of very nice UI's.

Ionic allows developers, with web experience, create mobile apps easily. When created they can be compiled using PhoneGap or Cordova before being pushed to the relevant store[11].

It allows the use of over 120 native device features such as Bluetooth and Finger Print Authentication. There are also Cordova and PhoneGap plugins available[12].

5.4 PhoneGap

PhoneGap is a framework by Adobe allowing cross-platform development of mobile apps using a single codebase. The applications will have access to native phone services, ensuring they function like native apps[13]. Web technologies HTML5, CSS and JavaScript are used to develop the mobile applications.

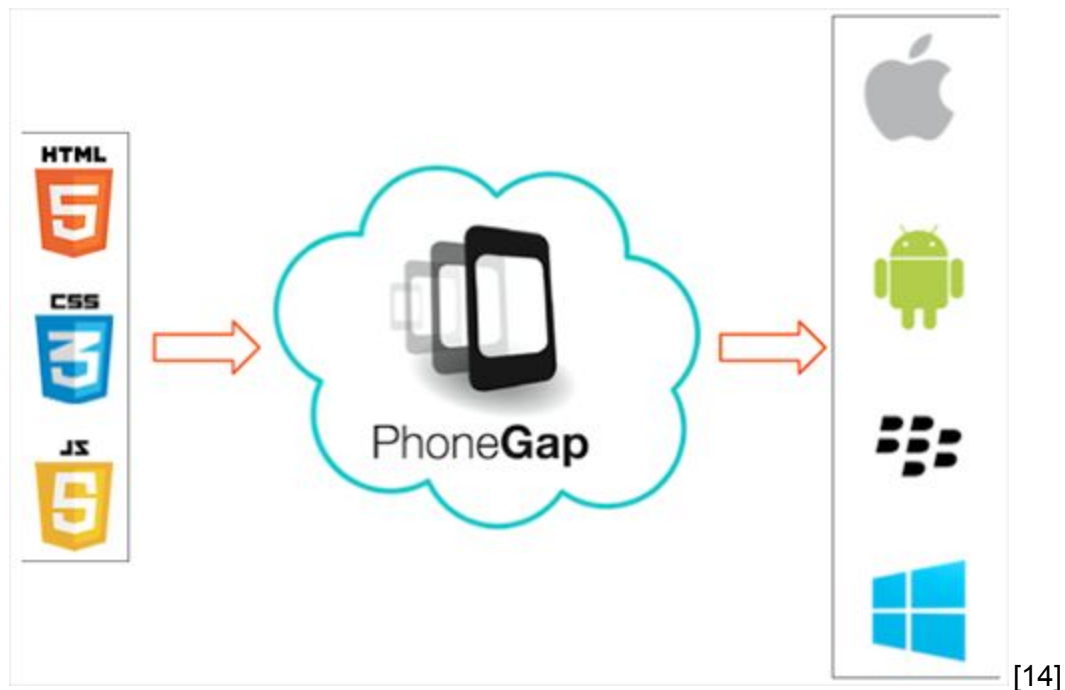


Figure 5.4.1 PhonGap

PhoneGap is a distribution of Apache Cordova.

6. Related Systems for SportsPA

6.1 Hudl

Hudl was founded in 2006 in Lincoln in the U.S.A. It has now expanded worldwide with offices in London and Sydney. Since its foundation it has grown rapidly and has acquired a lot of its competitors in the market. Two of its biggest competitors were Digital Sports Video (DSV) and APEX Sports Software in 2011 and 2012 respectively. It continued to make acquisitions of its biggest competitors in the following years with the purchase of Replay Analysis which was based in London and Sportstec in Sydney[24].

Hudl was initially designed to accommodate both college and professional football teams in America, but has expanded to work with many colleges and sports teams worldwide. It provides tools and software that allows coaches, managers and athletes to record game footage and events. It provides a platform where by recorded events and game footage can be reviewed and assessed. After video footage has been recorded of events or opponents, the video can be broken down. Voice clips and text feedback can be provided to coaches and athletes. This can lead to informed decisions being made which can help improve results[25].

6.1.1 Sportscodes

Sportscodes provides different packages designed to meet different packages and budgets.

The first offering from the Sportscodes package is Gamebreaker.

“Designed for teams looking for basic offline video review without the need for advanced statistical data[26].”

It allows the user to capture video either through a camera or a media feed. The captured video can be reviewed while recording is still taking place, without having to pause the video. The video can also be replayed from different angles. The user can code buttons for the recording of KPIs which are relevant and important, and will have full control over their design. When recording events using the coded buttons, lead and lag times can be set to ensure all of the relevant event is captured. Notes on important events can also be added. The recorded video can be exported for use outside of Gamebreaker+.

The next tool provided as part of the Sportscodes package is Sportscodes Pro. This offers the same as Gamebreaker+, along with extra functionality.

“Sportscode Pro allows teams to develop better workflows to capture, review and present video and analysis[26].”

More information can be captured by providing meta-descriptions with labels. Calculations on the data captured can be done while the coding is taking place. Presentation tools are provided that use the playlists recorded to replay events.

The top of the range of Sportscode is Sportscode Elite.

“Extract the most insight out of your video and statistical analysis with the software preferred by the world's top professional teams[26].”

7. Python libraries for parsing XML

In order to be able to use the data provided in the xml file, this file will need to be parsed into a format which will be usable when creating charts and plots. There are a few python libraries available to achieve this and the two most popular have been researched before making a decision on which to use.

7.1 `xml.etree.ElementTree`

“The ElementTree library provides an Element type, which is a simple but flexible container object, designed to store hierarchical data structures such as simplified XML infosets. The library also provides support for reading and writing Element structures as XML[15].”

ElementTree uses a python API and is implemented using the C language. It is three times faster than any other python package used for parsing and creating XML files[16]. Due to the hierarchical nature of XML files ElementTree would be very suitable for parsing and creating XML files.

This ElementTree hierarchical container contains two classes. The first class ElementTree is a representation of the full XML document. The second class Element represents each single node in the tree. The root Element of the Element Tree can be found by using the parse function on the Element tree. The parameter used in the parse function will be the XML file, eg root =

`ElementTree.parse(XMLfile)`. This will give the root node in the Element tree. The function `'find()'` can then be used to find the nodes at the next level in the tree. Each node will have two attributes which are the tag and the data. The `'iterfind()'` function iterates over a level to find all child nodes of that level and will return the tags. In order to retrieve the data of a particular tag the `'findtag()'` function can be used.

The above functions can be carried out on any XML file to access any of its data. It is also possible to change data in the XML file by accessing the data as outlined above. The changes can then be written by using the ElementTree `'write()'` function. New child elements are created by using `'SubElement()'`.

An example of an xml file to be parsed can be seen in Figure 7.1.1.

```
<data>
  <items>
    <item name="expertise1">SQL</item>
    <item name="expertise2">Python</item>
  </items>
</data>
```

[22]

Figure 7.1.1

In order to access tags and data in this file, we must first access the root of the parsed file. This can be done as shown in Figure 7.1.2. This will allow access to all nodes via the root.

```
root = tree.getroot()
```

[22]

Figure 7.1.2

To access the data in each of the nodes in the document we can use the code as shown in Figure 7.1.3. The code first iterates through all the immediate children of the root. It then finds all the nodes at the next level in the tree and prints the results.

```
for elem in root:
    for subelem in elem:
        print(subelem.text)
```

[22]

Figure 7.1.3

The resulting output will be as shown in Figure 7.1.4.

```
SQL
Python
```

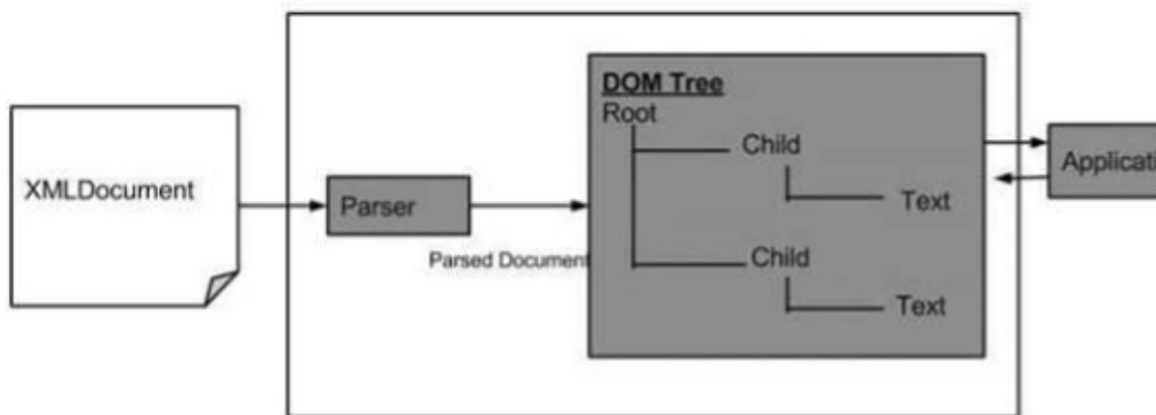
[22]

Figure 7.1.4

7.2 xml.dom.minidom

The Document Object Model, which is a W3C standard, allows programs and scripts to dynamically access and update documents, by providing an interface which is platform and language neutral[20]. With regard to xml specifically, it is an API for accessing, manipulating and changing xml documents. An implementation of the DOM, builds a hierarchical tree structure which is a representation the the xml file. It can also be used to create and build an xml document by adding tags and the corresponding text or data at the appropriate level of the tree structure.

An xml file can be loaded and will be parsed to produce a DOM representation of the document. This structure can be seen in Figure 7.2.1.



[21]

Figure 8.2.1

When the document has been parsed into a DOM representation, any or all relevant information can be retrieved, or added, by accessing the DOM. To achieve this, each level of the tree through the root of the tree which is the documentElement. This documentElement then has childNodes which represent the branches of the tree. An example of an XML file to be parsed into a DOM representation can be seen in Figure 7.2.2.

```

<?xml version="1.0" encoding="UTF-8"?>
<bookstore>

  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>

  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>

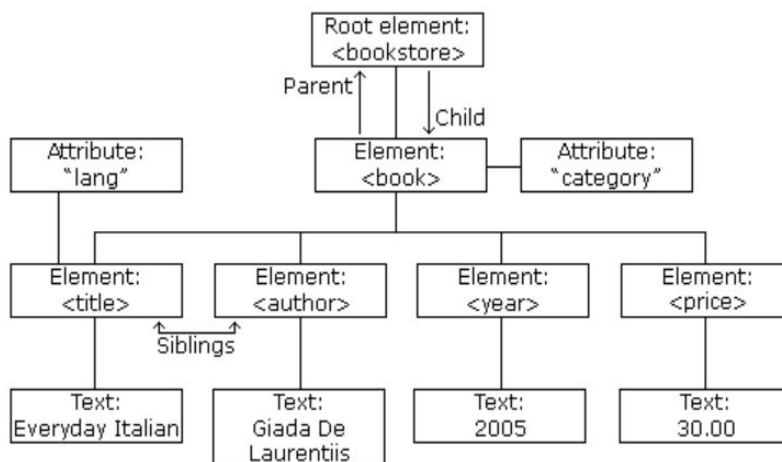
</bookstore>

```

[21]

Figure 7.2.2

The above XML file, after parsing into a DOM representation can be seen in Figure 7.2.3.



[21]

Figure 7.2.3.

In the above example the Root Element of the tree is the bookstore. Each parent node can have many child nodes and in turn each child can become a parent and have its own child nodes.

Each node can also have other nodes called attribute node. Text nodes then hold data relating to that node. Leaf nodes are nodes that do not have any child nodes. Nodes that are on the same level as other nodes, and have the same parent are sibling nodes.

The python library `xml.dom.minidom` is a minimal, and simpler, implementation implementation of the DOM[23]. It can be used to parse and manipulate xml files, using the DOM as described previously. The `minidom parse` function loads the xml files and parses it into a document variable. This document variable will be a DOM representation of the xml file.

There are different ways to access each node from the document. The `nodeName` and `tagName` attributes could be used as we move through the tree. The tree can be scaled by moving through nodes such as `firstChild`. Another way to access nodes is to use the `getElementByTagName()` function. Passing the name of the tag to be accessed, as a parameter into this function, will return the details of all tags with that name. Iterating over nodes, using the function `iterFind()` will allow the user to access all tags that are required.

7.3 Conclusion

This section of the project research involved looking at both `ElementTree` and `minidom`, with a view to using one of them to parse the xml files being used to store events. They are both very similar in that they both use a hierarchical tree structure to build a model of the xml file. The `minidom` library is the oldest and is a W3C standard. The `ElementTree` is far newer and was developed to be more pythonic in its structure. With both being so similar the decision was made to use the `elementTree` purely because of its pythonic nature, seen as python was the chosen language to develop the project in.

8. Python Visualization

8.1 Bokeh

Bokeh is a python data visualisation library, created by Continuum Analytics which creates interactive visualisations. The visualizations can be output as Json objects, as well as in html documents and interactive web applications[17]. The real strength of Bokeh is in its ability to stream and real time data, as it is dynamic and data driven.

Bokeh allows for front end interactions using python rather than JavaScript. It uses the canvas to render visualisations so it can handle a lot of data points. The use of canvas means that CSS will not be needed in general, except for some tools and widgets. Data visualizations on the bokeh server allow for real time updates. Any time the data changes the visualization will change in real time. Any complex computations can be done on the server side using bokeh server rather than in the browser. The bokeh server allows for complex data manipulation and the live streaming of data.

The python code is written on the server side and is turned into json objects. This json file is then sent to the client side, where a bokehjs library uses javascript to turn the json into native web applications using canvas and html. It is also possible to use other languages, such as R, scala and lua to create the JSON file which will also use the bokehjs library to convert the json. Visualizations which have already been rendered using other python libraries, such as matplotlib, pandas and seaborn, can also be rendered on the web using bokeh.

Bokeh has 3 different interface levels which can be used to create visualizations. A high level layer which uses charts, a mid level layer which uses plotting and a low level layer. The charts layer allows for the quick and easy creation of common charts such as histograms and bar plots. The middle level allows the user to gradually build up the plot. An example would be the dots in a scatterplot. The lowest requires some expertise as every element in the plot must be specifically defined[17].

Although Bokeh allows for the use of python for interactions, there are times when the python Bokeh library can't achieve what is required. For these instances, Bokeh allows the user to use custom javascript to achieve what is required. The custom javascript, known as CustomJS, is provided using callback functions. By using these CustomJS callbacks, it is possible to deal with interactions without using a Bokeh Server.

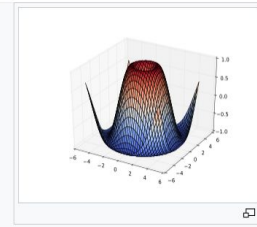
8.1 Matplotlib

Matplotlib is the oldest of the python visualization libraries and was based on the proprietary programming library Matlab. It is used for making plots of arrays. It can easily handle very large arrays of data through its use of the python library NumPy[18]. Many of today's python visualization libraries are built on top of matplotlib. Libraries such as pandas and Seaborn are wrappers around matplotlib and allow the user the benefits of matplotlib with less coding[17].

When Matplotlib was first released in 2003, it was designed with the view that simple plots should be created without the need to instantiate objects or call methods[18]. An example of a 3D plot created with just a few lines of code can be seen in Figure 8.1.1.

3D plot

```
>>> from matplotlib import cm
>>> from mpl_toolkits.mplot3d import Axes3D
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> fig = plt.figure()
>>> ax = fig.gca(projection='3d')
>>> X = np.arange(-5, 5, 0.25)
>>> Y = np.arange(-5, 5, 0.25)
>>> X, Y = np.meshgrid(X, Y)
>>> R = np.sqrt(X**2 + Y**2)
>>> Z = np.sin(R)
>>> surf = ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=cm.coolwarm)
>>> plt.show()
```



[19]

Figure 8.1.1

The code used to plot in Matplotlib can be broken down into 3 parts. The first part is the pylab interface. This interface provides functions from matplotlib.pylab which allow the user to write code to create plots. The second part is the Matplotlib API. This front end does all the work in creating the desired plot by creating glyphs such as lines and dots. The API does the work without knowing anything about how the plot is displayed as output. The third part is the backend which does the drawing. The backends are device dependent and an example of such a backend is Scalable Vector Graphics, SVG, which renders the plot.

9. Conclusion

A lot of research was carried out in the first 3 weeks which related to my original idea and then to the first idea proposed by Johnny Bradley who lectures in Sports Performance Analysis at IT Carlow.

The only decisions which really had to be made with regard to the SportsPA project was what python libraries to use. The first decision taken was to use ElementTree for parsing the data. This decision was taken purely from experimentation I had done on the libraries researched. ElementTree was the library I gained most familiarity with and seemed most suitable for my requirements.

The second decision was what visualization library to use. I had already decided to use Bokeh for the first proposal Johnny put forward. The reason for this was because this proposal required real time visualization and Bokeh was a new library designed with this in mind. When my project changed I stuck with Bokeh for visualization. I had already gained some experience, though quiet minimal, and was unsure if real time visualization might become a part of SportsPA.

10. References

[1] F. John Reh. 2017. *The Basics of Key Performance Indicators (KPI)*. [ONLINE] Available at: <https://www.thebalance.com/key-performance-indicators-2275156>. [Accessed 19 February 2018].

[2] performasports. 2018. *Setup your KPIs*. [ONLINE] Available at: <https://performasports.com/gaelic-football>. [Accessed 19 February 2018].

[3] staltz. 2018. *The introduction to Reactive Programming you've been missing*. [ONLINE] Available at: <https://gist.github.com/staltz/868e7e9bc2a7b8c1f754>. [Accessed 19 February 2018].

[4] codelabs. 2018. *Building Beautiful UIs with Flutter*. [ONLINE] Available at: <https://codelabs.developers.google.com/codelabs/flutter/#0>. [Accessed 19 February 2018].

[6] staltz. 2018. *The introduction to Reactive Programming you've been missing*. [ONLINE] Available at: <https://gist.github.com/staltz/868e7e9bc2a7b8c1f754>. [Accessed 19 February 2018].

[7]Bram Vanbilsen. 2017. *Flutter SDK Tutorial - Stateless Widgets! (App Development)*. [ONLINE] Available at: <https://www.youtube.com/watch?v=6f1rTeDm9oE>. [Accessed 19 February 2018].

[8] Frederic Lardinois. 2016. *Google's Dart programming language returns to the spotlight*. [ONLINE] Available at: <https://techcrunch.com/2016/10/26/googles-dart-programming-language-returns-to-the-spotlight/>. [Accessed 19 February 2018].

[9] firebase.google.com. 2017. *Firebase helps you build better mobile apps and grow your business..* [ONLINE] Available at: <https://firebase.google.com/>. [Accessed 31 October 2017].

[10] Google I/O. 2016. *FireBase Overview*. [ONLINE] Available at: https://video.search.yahoo.com/search/video;_ylt=AwrC5rkuPhZxA0AxfX7w8QF;_ylu=X3oDMTBncGdyMzQ0BHNIYwNzZWfYy2gEdnRpZAM-;_ylc=X1MDOTY3ODEzMDcEX3IDMgRhY3RuA2NsawRiY2sDNGtjamphbGNucm9tZyUyNmllM0QzJTI2cyUzRGNzBGNzcmNwdmlkA2lSLnpzREV3TGpGS01uTIZXWdNpMEFpSk1UUTVMZ0FBQUFCbnpBUXIEZnIDbWNhZmVIBGZyMgNzYS1ncARncHJpZANnYXk3Uy5adIRrcUxXbkZGME0uYThBBG10ZXN0aWQDbnVsbARuX3Jz bHQDNjAEbl9zdWdnAzEEb3JpZ2luA3ZpZGVvLnNlYXJjaC55YWhvby5jb20EcG9zAzEEcHFzdHIDZ29vZ2xlIGZpcmViYXNIBHBxc3RybAMxNQRxc3RybAMyMwRxdWVyeQNnb29nbGUgaW8gMjAxNiBmaXJlYmFzZQR0X3N0bXADMTUwOTQ3MjM5MQR2dGVzdGkA251bGw-?gclid=ga y7S.ZvTkqLWnFF0M.a8A&pvid=iR.zsDEwLjFKMnNVWX3i0AOJMTQ5LgAAAABnzAQR&p=goo

[gle+io+2016+firebase&ei=UTF-8&fr2=p%3As%2Cv%3Av%2Cm%3Asa&fr=mcafee#id=1&vid=19f86a814596bdb3783f8dfa4d946f3&action=view](https://www.google.com/search?q=firebase+google+io+2016+firebase&ei=UTF-8&fr2=p%3As%2Cv%3Av%2Cm%3Asa&fr=mcafee#id=1&vid=19f86a814596bdb3783f8dfa4d946f3&action=view). [Accessed 19 February 2018].

[11] firebase.google.com. 2017. *Firebase helps you build better mobile apps and grow your business.*. [ONLINE] Available at: <https://firebase.google.com/>. [Accessed 31 October 2017].

[12] Ionic. 2017. *The top open source framework for building amazing mobile apps.*. [ONLINE] Available at: <http://ionicframework.com/>. [Accessed 1 November 2017].

[13] Radar. 2013. *PhoneGap basics: What it is and what it can do for mobile developers.* [ONLINE] Available at: <http://radar.oreilly.com/2011/10/phonegap-mobile-development.html>. [Accessed 1 November 2017].

[14] tutorialspoint. 2013. *PhoneGap - Overview.* [ONLINE] Available at: https://www.tutorialspoint.com/phonegap/phonegap_overview.htm. [Accessed 1 November 2017].

[15] Stefan Behnel. 2011. *ElementTree.* [ONLINE] Available at: <https://wiki.python.org/moin/ElementTree>. [Accessed 19 February 2018].

[16] thegreenplace. 2012. *Processing XML in Python with ElementTree.* [ONLINE] Available at: <https://eli.thegreenplace.net/2012/03/15/processing-xml-in-python-with-elementtree/>. [Accessed 19 February 2018].

[17] Melissa Bierly. 2016. *10 Useful Python Data Visualization Libraries for Any Discipline.* [ONLINE] Available at: <https://blog.modeanalytics.com/python-data-visualization-libraries/>. [Accessed 19 February 2018].

[18] John Hunter. 2018. *matplotlib.* [ONLINE] Available at: <https://matplotlib.org/users/history.html>. [Accessed 19 February 2018].

[19] Wikipedia. 2018. *matplotlib.* [ONLINE] Available at: <https://en.wikipedia.org/wiki/Matplotlib>. [Accessed 19 February 2018].

[20] W3C. 2018. *Document Object Model (DOM).* [ONLINE] Available at: <https://www.w3.org/DOM/>. [Accessed 19 February 2018].

[21] Tutorialspoint.com. 2018. *XML DOM.* [ONLINE] Available at: https://www.tutorialspoint.com/dom/xml_dom_overview.htm. [Accessed 20 February 2018].

[22] Guru99.com. 2018. *Python XML Parser Tutorial: Read xml file example(Minidom, ElementTree).* [ONLINE] Available at: <https://www.guru99.com/manipulating-xml-with-python.html>. [Accessed 20 February 2018].

[23] python.org. 2018. *19.9. xml.dom.minidom — Minimal DOM implementation*. [ONLINE] Available at: <https://docs.python.org/2/library/xml.dom.minidom.html>. [Accessed 20 February 2018].

[24] Wikipedia. 2018. *Hudl*. [ONLINE] Available at: <https://en.wikipedia.org/wiki/Hudl>. [Accessed 20 February 2018].

[25] crunchbase. 2018. *Hudl*. [ONLINE] Available at: <https://www.crunchbase.com/organization/hudl#section-overview>. [Accessed 20 February 2018].

[26] hudl. 2018. *Compare Products*. [ONLINE] Available at: <https://www.hudl.com/elite/product-comparison>. [Accessed 20 February 2018].