“A Web-based GPS Data Visualisation tool for Sport Scientists”

Kanyakorn Charoensuk
H00172678

August 2014

Computer Science
School of Mathematical and Computer Sciences

Dissertation submitted as part of the requirements for the award of the degree of MSc in Information Technology (Business)
DECLARATION

I, Kanyakorn Charoensuk confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged. A list of the references employed are included.

Signed: ..................................................................

Date: .............................................................................
Abstract

GPS data is increasingly being used by sport teams for training, however for most sports clubs, teams and athletes, there are few affordable tools which enable the visualization of the data. Generally, they use simple excel functions to work out how strenuous the match or exercise was for each player. For example, they work out how many minutes were at 5km/hr, how many minutes at 10km/hr, etc. This research aims to demonstrate a tool that visualises and displays sports GPS data with the aim of enabling a new company to gain investment to develop the tool commercially.

Most current tools cannot support visualisation of data in a way that is helpful to sports coaches and athletes. There is therefore a gap in the market for cheap web-based tools where GPS data can be uploaded and useful information can be displayed. In the sport software market, some cheap alternatives exist, but they are typically not sports-specific and difficult to use. Sports analysis software solutions that do currently exist are expensive because they usually come with large multi-functional products such as ‘The Sports Office’. Though these can visualise the data in various ways, and display features such as speed, time spent in different parts of the pitch, etc. They are often too expensive for most sports clubs. Also, existing tools make it difficult or impossible for the sports scientist to configure the details of what can be visualised.

In this project, development of a GPS data visualisation tool that aims to be easy to use, with the ability to visualise basic aspects of the athlete’s performance, and also with the ability to configure aspects of what is to be visualised (e.g. details of how to divide the pitch into areas). After producing a demonstration of this application, it will be evaluated by Sports scientists at HWU to gain feedback and suggestions by electronic questionnaire. This project will help computer scientists or developers to see the capabilities of HTML5, JavaScript, etc. in this context. It will also help them in testing certain ideas about what to visualise, what graphs to show, etc. If this proves successful it is hoped that investment will be attracted to a consortium of HWU computer scientists, sports scientists, and especially ‘High Performance Sports Innovation Ltd’, which is a new company started by the above consortium in order to commercialise the product.
Acknowledgement

This project is not complete without showing gratitude to those that made the project a success. Firstly, thank you to the University for providing a conducive learning environment including the library facility which assisted me immensely in the literature review. And I want to appreciate all my instructors for their effort in exposing me to courses which I undertook during the entire programme. Thankfulness to my Supervisor Professor David Wolfe Corne who guided me and gave me encouragement to carry on with the project to its completion. Finally, my parents and my aunt for their moral, financial and spiritual support all through the commencement of the programme.
Table of Contents

Chapter 1 ........................................................................................................... 8
  Aims of the project ......................................................................................... 8
  Objectives ...................................................................................................... 8
  Introduction ................................................................................................... 9

Chapter 2 ........................................................................................................... 11
  Literature Review ........................................................................................ 11
  2.1 Overview of the Literature review ............................................................ 11
  2.2 Sport data visualisation tools, especially GPS or location ....................... 12
  2.3 Basic GPS data visualisation .................................................................. 15
  2.4 A web-based system or a non-web application ........................................ 16
  2.6 Web technologies for uploading and working with data files .................. 20
  2.7 Basic security issues for web-based application ....................................... 24
  2.8 Conclusion .............................................................................................. 27

Chapter 3 ........................................................................................................... 28
  Requirements Analysis .................................................................................. 28
  3.1 Overview of Requirements Analysis ......................................................... 28
  3.2 Functional Requirements ....................................................................... 28
  3.3 Non-Functional Requirements .................................................................. 30
  3.4 Professional, Legal, and Ethical issues ..................................................... 31
    3.4.1 Professional Issue ............................................................................. 31
    3.4.2 Legal Issue ....................................................................................... 32
    3.4.3 Ethical Issue ..................................................................................... 32

Chapter 4 ........................................................................................................... 33
  Project Plan and Risk Management ............................................................... 33
  4.1 A Project Plan (tasks, timeline, and Gantt chart) ....................................... 33
  4.2 Risk Management Plan ........................................................................... 34
  4.3 Conclusion .............................................................................................. 36

Chapter 5 ........................................................................................................... 37
  Design and Implementation .......................................................................... 37
  5.1 Overview of design and implementation .................................................. 37
  5.2 System structure chart ............................................................................ 38
    5.2.1 User registration ............................................................................. 39
Chapter 6
Testing and Evaluation

6.1 Overview of testing and evaluation
6.2 Testing

6.2.1 Component Testing
6.2.2 Integration Testing
6.2.3 Usability Testing

6.3 Evaluation
6.4 Conclusion

Chapter 7
Achievement and Limitation

7.1 Overview of Achievement and Limitation
7.2 Achievements
7.3 Limitation and Future Work

Chapter 8
Conclusion
List of Figures

Figure 1: I-gotU tool and Endomondo software ................................................................. 12
Figure 2: Sample interface of QSport software ................................................................. 13
Figure 3: thesportsoffice diagram ............................................................................... 14
Figure 4: Example of alternative chart on Google Chart Tools ........................................ 20
Figure 5: Example of security of web application architecture ........................................ 25
Figure 6: Gantt chart for project plan ............................................................................. 33
Figure 7: Structure Chart for a web-based GPS data visualisation tool for sport scientists system ................................................................. 38
Figure 8: This figure represents the line of code in home.php to be processed after the user uploads GPS files................................................................. 44
Figure 9: The example code to shows transfer from user directory to new location .......... 44
Figure 10: Example of code for reading file uploaded from user and formulate the data in well form .45
Figure 11: The example of .csv file............................................................................. 46
Figure 12: Interface of uploading file function ................................................................ 46
Figure 13: HTML code to call canvas .......................................................................... 47
Figure 14: JavaScript code to draw visualised pitch on the web application from actual data ......................................................................................... 47
Figure 15: The actual coordination from the pitch in Heriot Watt University .......... 48
Figure 16: The lines of code to show rotateCoordinates() function including code for indicating value of coordinate .............................................................................. 48
Figure 17: The rotate() function ................................................................................ 49
Figure 18: The example interface of players’ visualisation on an application ................. 50
Figure 19: The drawPlayer() function to visualise each player with specific number .......... 51
Figure 20: An example interface of player details ........................................................ 51
Figure 21: The code represents how to extract the number of players and their names .......... 52
Figure 22: The 355th and 356th line in the code to set the default value of number of rows and columns ................................................................. 53
Figure 23: Interface of an application for user to input number of row and column .......... 53
Figure 24: changeRowColumn() function will execute when user inputs a new number of row and column ......................................................................................... 54
Figure 25: An interface of time spent of each player in particular area ..................................... 54
Figure 26: calculateColourCodeGrid() function ........................................................... 55
Figure 27: drawColourCodeGrid() function to colour the pitch ..................................... 55
Figure 28: An interface of drop box for player selection .................................................. 56
Figure 29: createDropdown() function ........................................................................ 57
Figure 30: The function drawChart() is used to visualise the speed and time graph .......... 57
Figure 31: The function drawChart2() is used to visualise the speed and time graph ....... 58
A Web-based GPS Data Visualisation tool for Sport Scientists

Figure 32: An interface of speed over time graph .................................................................58
Figure 33: An interface of distance over time graph ..........................................................59
Figure 34: An interface of registration page ........................................................................60
Figure 35: An example code to ensure new user input the required information ..............60

List of Tables

Table 1: The performance of Flash and JavaScript on web-based application ..................19
Table 2: show 4 functions for FileReader ............................................................................23
Table 3: Requirements' checklist .........................................................................................30
Table 4: Risk Plan Table .....................................................................................................35
Chapter 1

Aims of the project

The aim of this project is to develop a prototype of a sports GPS data visualisation and analysis system to help a new company which is called High Performance Sport Innovation Ltd (http://sportinnovation.co.uk/) to gain investment to develop the tool commercially and the tool will also be used within HWU by the National Performance Centre for Sport (which is part of Heriot Watt University). Basically, there are two main ways that this project can benefit the new company. Firstly, the company will be able to see the potential functionality and the possibility of developing a commercial web-based tool. For this purpose, this project helps sport scientists see the capabilities of web development tools such as HTML5, JavaScript, and so on. It will also help a company in testing certain ideas about what to visualise, what graphs to show, etc. Secondly, this project could be a demonstration version that could be shown to investors.

Objectives

For this project, the main objectives are:

- To produce a demonstration of a tool that visualises and displays sports GPS data
- To demonstrate the potential advantages of this application for both sport scientists and initial business company
- To ensure the system is user-friendly
- To ensure the system is extendible and modifiable by non-experts
Introduction

Men and women of all ages the world over have recently shown an increased interest in taking up sports. An increase in leisure time gives people the opportunity to play sport to improve their health and to have fun. Many people seeking to improve their health and fitness will look for technological devices to help them to improve their performance. Most sports clubs have modern sports equipment and devices, including GPS data tracking devices. These devices are used to time and record running speed on a 5km run for instance. Most analysis by sports scientists is done by analysing videos of the event. The analyses usually aim to extract summary data about the player's performance or effort, how they moved around the playing surface, and how these details changed over time during the game. GPS data is quite recent in sports, and sports scientists expect that it will help them in doing these analyses. However, the current tools available to them to make use of the GPS data are not affordable and/or ineffective. What such users want is a technological device with which to visualise the relevant data.

The main purpose of this project is to demonstrate a tool that visualises and displays sports GPS data with the aim of enabling a new company such as High Performance Sport Innovation Ltd (http://sportinnovation.co.uk/) to gain investment to develop the tool commercially and this tool will also be used within HWU by the National Performance Centre for Sport (which is part of Heriot Watt University) for sport scientists wishing to visualise data.

A web-based GPS Data Visualisation tool will help guide sport scientists in the use of web tools developed with the aim of visualising GPS data. This paper starts by introducing sport tools and the concept of GPS data visualisation and then investigating the appropriate tools for this application in chapter two. Chapter three will state the requirement gathering phase in the software development life cycle is a vital phase as it will ensure that this project meets all user needs. In order to emphasise that the software developed will meet such needs, information on both functional and non-functional requirements was gathered directly from sport scientists. In chapter four, a project plan and risk management plan was explained including identifying the critical phase during software development process. Next, a
description of software design and how to develop web applications to work with single and multiples files and visualise the GPS data is also given in chapter five.

In chapter six, three testing strategies were applied in this project, namely component, integration, and usability, in order to ensure that this application possesses the qualities of accessibility, reliability and usability. A questionnaire was administered to elicit feedback and suggestions for improvement. Chapter seven described the achievements, limitation, and future work. Finally, the conclusion of this project with the multi-dimensional advantages are being presented in chapter eight which will provide readers with a clear and detailed picture of this project.
Chapter 2

Literature Review

2.1 Overview of the Literature review

Technology plays an increasingly important role in sport nowadays and can significantly help people improve their performance. It is used with specialised equipment and State-of-the-art Technologies to perform tasks more efficiently. Examples of sporting technologies include clothing, footwear, athletic sports gear, advanced computer simulations and motion capture, etc. However, the software tends to be very costly. This project will produce a demonstration of a tool that visualises and displays sports GPS data with the aim of helping a new company gain investment to develop the tool commercially. This tool will allow the user to input multiple GPS data files into the system and visualise them in terms of a grid showing players’ movements or speed. In this research possible tools for constructing the application to achieve the objectives of this project are investigated, with consideration to which is the most appropriate technology for visualising GPS data over the web. The following literature reviews attempt to demonstrate and support this proposal.

These literature reviews help expand web application knowledge for this project. It starts with an introduction to existing sport data visualisation tools in section 2.2. Then, basic GPS data visualisation in section 2.3. In section 2.4, a web-based system or a non-web application are discussed. Data visualisation on web browser tools are investigated in section 2.5 and web technologies for uploading and working with multiple data files in section 2.6. The last section (2.7) provides an analysis of security issues for a web-based application.
2.2 Sport data visualisation tools, especially GPS or location

In recent years, sport software has become popular and has produced a paradigm shift in sport science. It is used by those concerned with fitness, health and improving athletic performance. An athlete’s health can be improved, observed and injuries treated, through the production of sports technology such as heart rate monitors, pedometers and body-fat monitors, sport gear (clothing and foot wear), sporting equipment such as the composite tennis racket has been created for providing enhanced ball speed, also bicycles have seen the development of specialist wheels, pneumatic tyres, break levers and pedals, which will improve stability and rigidity of the bicycle. Giannakis, Chorianopoulos, Jaccheri (2013) investigate what are the user requirements for sports tracking software. This technology includes GPS sensors that provide speed, distance, elapsed time, and mapping. The first device considered is I-gotU, it is a small GPS logger that is a stand-alone portable device. It is of limited use as it does not have any screen or speaker. By contrast another tool is the Endomondo, users will get the real-time results along with map visualisation. In a questionnaire, both GPS technology devices were considered safe, comfortable and handy. The I-gotU was considered mediocre whereas the Endomondo got a positive ranking. This suggests that people prefer devices that have more interaction and visualisation.

There are also wearable sensor systems which are used for visualisation of data called Visualisation in Team Scenarios or VTS (Kazmi, O'Grady and O'Hare, 2011). VTS can display physiological data to players such as heart rate, respiration rate, skin temperature and GPS data. A coach can then analyse the data to improve the performance of the athletes.
A prototype that has a base station connected to a laptop was implemented in Java. It processed in-motion graphics and complex data visualisations. From this research, sport science in the future will probably need to integrate a range of sensors in order to monitor and improve the athletes’ performance.

Another product in the sports software market for athletes with a limited budget is **QSports**. This provides sports management software with an integrated database, enabling athletes to:

- Show how much time and distance they work out each week
- Provide information about how many calories have been burned during exercises such as cycling.
- Store and analyse their workouts using various sports gear.
- View the data history and produce meaningful statistical analyses.
- Record various activities
- Customize Fitness and Workout Goals
- Provide multiple user access, so coaches can review sports and health management data for each team member
- Provide graphical and map views of the data enabling users to review workout data not only in figures but also with charts, maps and tables

![Figure 2: Sample interface of QSport software](image-url)
However Qsport software does not support sports specific interfaces and has a low level of security. Though this product is not costly for sport teams, it is not the ideal software solution to enable sport scientists and athletes to visualise the GPS data in a useful way.

The most effective sport software currently available to maximise performance is “thesportsoffice”. It offers sports analysis software for a range of sports including football, rugby, cricket, and other activities.

thesportsoffice main features are:

- A powerful centralised hub for all playing, coaching, performance, operations and medical personnel.
- Powerful information management, statistical analysis and data mining.
- Communication with players/athletes and between all departments.
- Instant timetabling and scheduling to improve organisation and logistics.
- Effective monitoring of athlete workload in training, preparation and competition.
- Supports personal improvement plans for players/athletes.
- Coordinates recruitment, retention and succession planning.
- Secure access to HR section for all admin data.
- Providing hosted web-based solution with no specialist IT knowledge needed.

![Figure 3: The thesportsoffice diagram](image-url)
This software comes with a mobile application compatible with all types of smartphones and tablets and offers the user the ability to access the system in any location, at any time. It provides a rich interface with powerful features, however it also comes at a high cost, consequently most clubs cannot afford to purchase it.

2.3 Basic GPS data visualisation

The visualisation software proposed in this research will process GPS data and display it in a web application. GPS (Global Positioning System) plots a user’s position on the earth’s surface using a receiver and decoding time signals from several satellites (Clarke, 2011). According to Kraak and Ormeling (2003), the map is one of the tools to represent the location or the attributes of objects or phenomena located on earth in terms of geospatial data. A popular way to display this GPS data is using a web-site enabling easy access across all devices connected to the World Wide Web and multi user access. In addition, using a web application as the GUI (Graphic User Interface) to display the geospatial data is not only easy to access but also to control user input by writing the interface via HTML, Java, or Tcl/Tk. Users can use simple mouse interactions to interface with the system and it can analyse the information without requiring complex SQL skills. Another property of the web is generating dynamic views to visualise the data when users perform spatial queries. In order to provide suitable security, the information retrieval system is controllable by Internet security standards such as firewalls, and using HTML or Java logins with usernames and passwords. It can be protected by the use of authorized access user, encryption techniques, and providing multi-level user accessibility (Pundt and Brinkkötter-Runde, 2000).

There are many possible programming languages which can represent data (Datos, 2013). For example, Java, Flash, HTML, JavaScript, PostgreSQL and its PostGIS geospatial extension, Ruby, Python, etc. Software visualisation tools can provide meaningful views and can describe quantities of data clearly (D'Ambros, Lanza, Lungu, and Robbes, 2011). Furthermore, the software visualisation tool as a web application is a great help for making the information available to wider audiences. In this research, a number of technological alternatives and tools which are available to implement visualisation on web application are discussed.
2.4 A web-based system or a non-web application

In recent years, the move from the non-web application to the web-based application has been gradual, both web based applications and non-web based applications have their advantages and disadvantages. From section 2.3, it shows that a software visualisation tool as a web application is accessibility for various users, so comparing the web-based system with non-web application are investigated in this section.

Today cyber threats or attacks are more robust and powerful than ever (citrix, 2013), so effective information security is important and it is significantly challenging to maintain. Traditional Desktop Based Applications have facilitated enterprises to perform dynamically with security function and quick to install, but it is limited performance by many factors such as after deliver, it requires computer experts or IT technical staff to maintain the system. Moreover, it has been limited by the hardware on which it can run and the operating systems supported. Therefore, the software has to provide versions for all types of operating system for instance DOS, Windows, MacOS, UNIX – Linux, and so on. Updating the applications must be applied by the user directly to their installation, and may require hardware upgrades or other changes. As a result they are difficult to maintain in the long term and can be costly if robust changes are needed the future.

Apart from Desktop Based Applications, non-web applications also include mobile applications, which are popular today. According to Hersch (2013), there are a majority of mobile and tablet users increasingly in every years. Consequently mobile applications are in high demand in the technology market. Even if, it is accessible and provides convenient services for users, it still has some weak points. On most mobile devices, the speed of services are slower than dial-up Internet access and lack Javascript and cookies, it does not support client-side scripting including Flash, pdf, or video sites, and similar software. In addition, mobile applications offer a small screen size, it is not suitable for this project because it is powerful with screen that can visualise data in details.
For this project, it seems like a web-based application will have more benefits for data visualisation. A number of the main benefits are

i) Convenience, it is easy to access any time anywhere for users.

ii) No installation or maintenance required.

iii) Multiple Platforms, it is compatible with most of the computer operating systems.

iv) No third party fees, for example the App Store commands 30% of revenue.

However, a web-based application comes up with many services for users, but is not without risks. It still has some disadvantages such as it can be vulnerable to attack from malicious users or organised criminals. In addition, a web-based system requires an internet connection. So, if users do not have Internet access or their Internet connectivity is slow, it will be unusable or affect the performance of the system. In conclusion, all platforms have both advantages and disadvantages, which are discussed above. So, this project could more gain benefits from a web-based application than non-web application.

2.5 Data visualisation on web browser tools

In a research article by D'Ambros, Lanza, Lungu, Robbes, and Romain (2011), a number of software alternatives are investigated to implement software visualisation on web application with rich graphical and interactive elements. They considered JavaScript, flash, Silverlight, and Java applets. Firstly, JavaScript and DHTML. JavaScript (Wilton and McPeak, 2010) is a computer interpreted language that is a series of instructions that tell the computer to do a wide variety tasks such as display text, move an image, or ask the user for more information and combine functional and prototypical paradigm. Interpreted languages are computer languages that use an interpreter to translate the code to machine code for the particular OS (Operating System). This means they are more platform independent than a compiled language (such as Visual C#) which requires a compiler and is usually limited to a specific OS.

DHTML stands for Dynamic Hyper Text Mark-up Language; it defines the layout of the webpage and can be programmed by JavaScript using the DOM (Document Object Model)
defined by the W3C. DOM is used for modifying the content and interacting with the web application dynamically. Prototype and jQuery are two of widespread libraries and array of frameworks which can build the graphical user interface including charting and interaction widgets. jQuery is “an open source JavaScript library that simplifies the interactions between an HTML document, or more precisely the Document Object Model, and JavaScript” (Asokan, 2013). Examples of frameworks are Dojo, script.aculo.us, Sprout Core, Mootools, Yahoo UI Library, and Google Web Toolkit. JavaScript also can work with HTML canvas tag, which allows several visualisation libraries to be created on top of HTML canvas such as Processing.js, Cake, Raphael, and InfoVis Toolkit. Particularly Processing.js is a programme that enables visualisations, animations, and interaction in JavaScript.

JavaScript therefore provides the visualisation in the web application with flexible data manipulation. In addition other tools such as Java applets, Flash, and Silverlight can be used. Java applets are Java applications that run in a web application through the Java Virtual Machine and provide interactive features to web browser that HTML cannot provide. These applets are compatible cross-platform, and do not have the problems of some other Java technologies such as Java Web Start which does not run in the web application. In the single run time environment, Flash and Microsoft Silverlight can integrate graphics, animations, and multimedia. Furthermore, Flash provides Flare as a library that can explore line chart, bar chart, maps, and bubble chart, however the weak point is when it integrates with the browser as it cannot communicate with the other information in the web page and so acts likes a “black box”.

To sum up, Both Flash and JavaScript have strong and weak points. Since JavaScript provide libraries such as jQuery and MooTools for supporting animation without reloading a web page, it has advantages over Flash. According to logicpool (2009), the main advantages of Flash’s are: its extensive features, 3D Capabilities, and the provision of more font options. However it is not compatible with all devices including iPhone, iPad and cell phones, there is increased development with cost, and typically large file sizes to download. Another weakness is the problem of users having older versions of the Flash Player. By contrast JavaScript is a smaller size, provides a lot of free professional quality programs, can add interactivity to web page elements and tags, and is compatible with more browsers including iPhone, cell phones, PS3, PSP. However though Javascript has many strong points, it still has some limitation such as not being as feature rich and complex features may not perform as quickly as flash. Also 3D features are limited and the source code is not protected.
There are other tools for graphic visualisation in web applications such as HTML5. According to Ding, Xian, Le, Shu, Zhang, and Zhu (2012) the researchers said that “HTML5 is an emerging technology” because it can be used to combine C, C++, Java, JavaScript, and .NET with HTML5 integrating server-side and In-Browser code. Although, execution time of JavaScript is slower than C/C++ in most cases by at least an order of magnitude. JavaScript plays a significant role in HTML5 applications as it can execute the application’s logic within the browser, and provide user interaction without the time delay of posting the page back to the web server.

In order to provide visualisation of the data using charts, Google API provides a tool kit which is easy to learn, well-documented, free and open source. However, Google Chart Tools are based on HTML5 and scalable-vector-graphics technology. As reported by Zhu (2012), Google Chart Tools support various chart types such as pie charts, scatter charts, gauge charts, geocharts, tables, treemaps, combo charts, line charts, bar charts, column charts, area charts, and candlestick charts. In addition, it allows user to click on a visual entity, and it will display more details of that entity as a small pop-up. Users can also combine charts using visualisation dashboards. Google Charts Tools are exposed by JavaScript, and it can also be coded with Java, but it needs to integrate with the Google Web Toolkit. For the data view, user can send a SQL query from the client-side programme to the database, and it will generate appropriate information that it returns in a data table. Although, Google Chart Tools are flexible, cross platform and user-friendly, they only support 2D charts and graph drawing is unavailable. Therefore there is limited to customisation of the provided graphs and it is difficult to create new type of charts.

<table>
<thead>
<tr>
<th>Application</th>
<th>Flash</th>
<th>JavaScript</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slide Show</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Form Validation</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dropdown Menus</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Tabbed Panels</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Popups &amp; Tooltips</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Expandable/Collapsible Elements</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Video/Audio Player</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Complex Animation</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3D</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Complex Multimedia</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Table 1: The performance of Flash and JavaScript on web-based application
2.6 Web technologies for uploading and working with data files

According to MacCaw (2011), compared to file access and manipulation in desktop applications, web browsers are limited but additional functionality can be provided by plug-in technologies such as Adobe Flash. **Plug-in is an add-on JavaScript programme which works in conjunction with jQuery in order to make certain tasks, effects, or features incredibly easy to add to a web page** (Asokan, 2013). HTML5 allows developers more control of files and blurs the boundaries between desktop and web. The browsers that support the new HTML5 file functionality are Firefox 3.6, Safari 6.0, Chrome 7.0, and Opera 11.1. Unfortunately, Internet Explorer does not yet support this standard for providing users the ability to input, drag/drop and upload files. Code to check whether the relevant objects are supported by the browser can be used such as:

```javascript
If (window.File && window.FileReader && window.FileList){
   //API supported
}
```
For transferring files to the web server, XMLHttpRequest Level 2 specification is able to upload files but is only supported by some browsers:

- Safari >= 5.0
- Firefox >= 4.0
- Chrome >= 7.0
- IE: no support
- Opera: no support

Another option to upload files is to use the function send(), or alternative function FormData. For an easily manipulatable interface FormData is recommended.

Example code of object:

```javascript
var formdata = new FormData($('form')[0]);

// You can add form data as string
formData.append("stringKey", "stringData");

// And even add file objects
formData.append("fileKey", file);
```

If using jQuery for Ajax request, setting processData option to false is required and the user must not set the Content-Type header because the browser will set it automatically.

Example code for setting processData option to false:

```javascript
jQuery.ajax({
    data: formData,
    processData: false,
    url: "http://example.com",
    type: "POST"
});
```
Example code for passing file directly to XHR by send() function:

```javascript
var req = new XMLHttpRequest();
    req.open("POST", "http://example.com", true);
    req.send(file);
```

Example code for setting processData option to false from using send() function:

```javascript
$.ajax({
    url: "http://example.com",
    type: "POST",
    success: function(){ /*...*/},
    processData: false,
    data: file
});
```

As a result of using FormData objects, and sending the upload multipart/FormData request, users will see jquery.upload.js in assets/ch07 folder. It is an abstract file that upload into simple $.upload(url, file) interface. For the uploadFile() function, it can send an Ajax request to server by uploading $.upload() function in jquery.upload.js.

Example code:

```javascript
var uploadFile = function(file){
    var element = $('div');
    element.text(file.fileName);

    var bar = $('div');
    element.append(bar);
    $("#progress").append(element);

    var onProgress = function(e){
        var per = Math.round((e.position / e.total) * 100)
        bar.progressbar({value: per});
    };

    var onSuccess = function(){
        element.text("complete");
        element.delay(1000).fade();
    };
};
```
Another option is to read the file asynchronously using FileReader. This does not lock the browser whilst the data is being transferred and provides a callback to the file reader instance when the file is ready. There are 4 functions for FileReader depending on which data format is returned:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`readAsBinaryString(Blob</td>
<td>File)`</td>
</tr>
<tr>
<td>`readAsDataURL(Blob</td>
<td>File)`</td>
</tr>
<tr>
<td>`readAsText(Blob</td>
<td>File, encoding='UTF-8')`</td>
</tr>
<tr>
<td>`readAsArrayBuffer(Blob</td>
<td>File)`</td>
</tr>
</tbody>
</table>

Table 2: show 4 functions for FileReader

In conclusion, jQuery is the one of a range of tools which are suitable for uploading and working with multiple files in a web application. It provides many functions that are useful to users and can be accessed by most modern browsers. It is therefore that this project can gain benefits from this tool and successfully implement it to accomplish the objectives of this application.
2.7 Basic security issues for web-based application

In recent years, there has been a significant increase in attacks on web sites by hackers and as many as 70% of web browser inadequately detect new vulnerabilities. This can have a severe effect on business involving data theft and operational disruptions (ThomasNet News, 2013). Web site security is therefore very important and companies are paying increasing attention to the widespread adoption of security protocols in web technologies.

According to ThomasNet News (2013), new technologies provide improving protection from attack. For example, HTML5 offers new features but is also leading the way in new code vulnerabilities and attackers still find their way via unauthorised access to the business's web infrastructure. The new Acunetix Web Vulnerability Scanner combines black box scanning techniques with feedback from sensors placed inside the source code before it is executed. This provides extra levels of vulnerability detection. AJAX (Microsoft technology using asynchronous JavaScript and XML) can also open the door for attackers (Wadlow, Thomas, Gorelik and Vlad, 2009). A web site contains the code that connects the pc with the web server and conducts a conversation with the server using security mechanisms integrated into the browser. Another weak point with Flash, JavaScript, and Java is that it allows programs written by unknown third parties into the browser.

Another popular attack is cross-site scripting or XSS, where hackers will embed code in the webpage which will steal cookie information from your browser and transfer it to their website. Web security is therefore very important for protecting personal information but effective implementation depends often on the size of business too (Kesh and Ramanujan, 2004). For example, the needs of a small organisation maybe less than a large company. Both small and large organizations are vulnerable to attack by viruses, however large companies tend to experience more Denial of Service (DOS) attacks than small companies.
The main data security within HTML5 is ensured by only allowing file access when the user manually selects it. This can be done by dragging the file onto the browser or selecting it from a file input dialog box. If JavaScript was able to read and write arbitrary files on the user’s PC this would be a major security flaw. Multiple files and file path information is provided using FileList Objects which treat them as an array of File objects (MacCaw, 2011). There are three attributes for file objects in HTML5 namely:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The file’s name as a read-only string</td>
</tr>
<tr>
<td>Size</td>
<td>The file’s size as a read-only integer</td>
</tr>
<tr>
<td>Type</td>
<td>The file’s MIME type as a read-only string, or an empty string (&quot;&quot;&quot;) if the type could not be determined</td>
</tr>
</tbody>
</table>
In Pathan’s view (inderscience, 2012), Web applications are an increasing security risk with functionality allowing users to interact with the backend database and providing a rich user interface enabling various tasks such as updating data, issuing queries, and extracting data. For all these operations, a crucial role is played by the design interface and structure of the websites. Also the design of the backend database has a huge impact on the security of the web application. Plenty of websites lack proper levels of security and are vulnerable to attack allowing illegitimate users access to the backend databases.

Weak security in a web applications design will allow crafted injection and malicious update in the backend database. This can cause a lot of damage and runs the risk of the theft of trusted users’ sensitive data. In some cases, the attackers can gain full control over the web application and totally destroy or damage the system. Therefore, the level of security built into a web application is a critical issue in order to protect the sensitive data of authorised users.
2.8 Conclusion

In conclusion, from my reading and research into GPS data visualisation tools there are three main issues.

Firstly, existing solutions for providing GPS visualisation to athletes are typically available in larger multi-functional products which are difficult for someone who is not a specialist in IT field to use and too expensive for most sport clubs. While some cheap alternatives exist, they do not support specific features which are helpful to coaches and athletes.

Secondly, GPS data visualisation functionality can be well provided through the web browser. It is a useful way to provide a rich graphical interface and in a form familiar to users and also supports multi user access allowing the data to be made available to a wider audience.

Finally, after investigating the performance of various tools, and comparing the advantages and disadvantages of each. The researcher believe that the most appropriate tools for this project are:

- JavaScript libraries
  - Processing.js will be used for visualised data on website.
  - jQuery for working and uploading multiple files on web application
- HTML5 will support some data visualisation and security protocols
  - Google Chart Tools will be used for creating charts or graphs and working with HTML5

It is also recommended that further research into the other basic tools for creating a web applications such as HTML, CSS, JavaScript, PHP, and SQL, is undertaken to identify how best to use each tool to generate the website.
Chapter 3
Requirements Analysis

3.1 Overview of Requirements Analysis

This chapter identifies what was agreed with the stakeholders in this case the stakeholder was a director of HPSI Ltd., including mandatory and optional requirements that improve the project in terms of interface, features, and performance of the system. In addition a requirements checklist table is provided to enable the reader to visualise the scope of project clearly. For optional requirements, these are divided into five terms: Availability and Accessibility, Reliability, Security, Usability and Applicability, and Performance. On the other hand, not only the features of an application are required, but also the issues about professional, legal, and ethical should be included in this project too.

3.2 Functional Requirements

The project will implement the following mandatory functional requirements in order to meet the users need for:

- **Browsing**
  The web application should be available in various browsers in order that users will have a choice of browser and platform. Examples of supported browser are Google Chrome, Firefox, and Safari. Unfortunately, this application uses HTML5 as a tool, so it currently will not be supported by Internet Explorer (IE).

- **Interface**
  The web-application should be user-friendly and provide support for sports scientists who are not computing experts. Furthermore, it should provide a simple attractive interface in order to give clear features and allow users to interact with all the functionality off this application easily.
Functions or features

Generally, the system should collect the GPS data from users and visualise the information in a rich GUI (Graphical User Interface) using for instance, graphs, charts, and maps. The detail of features which this application should provide, are shown in requirements’ checklist table.

Requirements’ Checklist

<table>
<thead>
<tr>
<th>#Requirement</th>
<th>Description</th>
<th>Priority</th>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>The system can upload multiple GPS data files</td>
<td>High</td>
<td>✓</td>
</tr>
<tr>
<td>R2</td>
<td>The pitch and each player can be visualised on the field</td>
<td>High</td>
<td>✓</td>
</tr>
<tr>
<td>R3</td>
<td>Visualise player by the circle with specific number according to the order of input file</td>
<td>High</td>
<td>✓</td>
</tr>
<tr>
<td>R4</td>
<td>Provide table to represent number for players associated with their name</td>
<td>High</td>
<td>✓</td>
</tr>
<tr>
<td>R5</td>
<td>Ability to animate all of the players (a player is shown as a circle on the screen) and see players’ moving automatically</td>
<td>High</td>
<td>✓</td>
</tr>
<tr>
<td>R6</td>
<td>Allow user to input the number of row and column by providing a simple ‘analysis’ button; when pressed, the field is divided base on the user input.</td>
<td>High</td>
<td>✓</td>
</tr>
<tr>
<td>R7</td>
<td>The grid is colour coded based on the time that player spent in that part</td>
<td>High</td>
<td>✓</td>
</tr>
</tbody>
</table>
A Web-based GPS Data Visualisation tool for Sport Scientists

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R8</td>
<td>Provide a drop down box to select the player, and his field is colour-coded based on that player</td>
<td>High</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>R9</td>
<td>Visualise a speed/time graph for each player</td>
<td>High</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>R10</td>
<td>Visualise a distance/time graph for each player</td>
<td>High</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>R11</td>
<td>The system security is appropriate with this application</td>
<td>High</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Requirements' checklist

3.3 Non-Functional Requirements

The non-functional requirements of this project are listed below along with a range of optional requirements:

- **Availability and Accessibility**
  The GPS data visualisation tool should be available and accessible any time when users request to use the system.

- **Reliability**
  The ability of the GPS data visualisation tool to deliver services as specified without fault.

- **Security**
  System security is very important for the GPS data visualisation tool, because it will have personal information. So, the system should ensure that the data is protected from unauthorised access.
  - The system shall allow only access to users who are authorised and identify themselves by using a login name and password.
• **Usability and Applicability**
  The design should prioritise the user-experience and ensure it is easy to use for those who are not programming experts.

• **Performance**
  The system should respond in a timely fashion to user interactions and specifically enable:
  - A GPS file to be loaded and processed in less than 5 seconds
  - Each analysis function to be achieved in less than 5 seconds

### 3.4 Professional, Legal, and Ethical issues

The successful project will give consideration to the professional, legal and ethical issues in order to ensure the efficiency of the project as well as other issues relating to the construction of the entire project.

#### 3.4.1 Professional Issue

The whole development life cycle of the project should be undertaken to a professional standard in order that it meets the common international standards in terms of safety, time, quality, and communication with concerned parties involved in the project.
3.4.2 Legal Issue

Consideration should be given to the international legal issues of protecting user’s privacy. Specifically with regard to regulations applied by different bodies such as organisation, government, university and so on. In this project the legal issues will focus on the information provided by athletes, sports coaches, and sport scientists. Any personal details will not be disclosed without permission and should be considered a legal issue under the terms and conditions of the sport centre at Heriot-Watt University.

3.4.3 Ethical Issue

The data files that will be used in this research contain personal information about players such as their names and movements. However they have given assent to the HWU sports scientists to use this data, and it will be used only for showing to sports scientists. In demonstrations of the project by HWU to others, anonymous names will be used instead. For the commercial version, the players could work out the general level of fitness and their information is very important for them to bet on future games. Therefore, the system needs to be secure and, consented to wear a GPS device by player. Then, the player has consented to their GPS data to be viewed by the sports scientist.
Chapter 4

Project Plan and Risk Management

4.1 A Project Plan (tasks, timeline, and Gantt chart)

Figure 6: Gantt chart for project plan
This project will take 85 days, it is divided into 5 main tasks as shown below:

- Requirement Gathering (7 days)
- Design (10 days)
- Implementation (61 days)
- Testing and Debugging (11 days)
- Report and Poster (11 days)

For the critical path is implementation phase, the reasons why this phase were spent longest time, because a web application for GPS Data Visualisation tool is developed in this phase including reading some textbooks and learning programming skills for data visualisation on web application. Especially, during software development process, there are some challenges can be occurred, so the researcher have to spend time to overcome any bugs and errors.

4.2 Risk Management Plan

Risk is an important part of the challenge for developers. Some risks occur rarely, but can impact the whole project. Therefore, good management and a plan of risk management will decrease any negative impact on the project. Developers should pay attention to it before starting the project and continue assessing it during the whole development life cycle. The project’s risk plan should be updated regularly and especially on reaching a new phase in order to manage unplanned risk that may have a wide effect on the project. Risks that have been identified are listed in the following table:-
<table>
<thead>
<tr>
<th>NO.</th>
<th>Potential risks</th>
<th>Evaluation (Impact)</th>
<th>Level occurrence</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Unavailability of regular meeting with supervisor or preoccupation of supervisor with his own work.</td>
<td>Medium</td>
<td>Low</td>
<td>Contact with other teachers who have similar knowledge about the project or keep knocking on his door and sending emails.</td>
</tr>
<tr>
<td>2.</td>
<td>Health problems</td>
<td>High</td>
<td>Low</td>
<td>Getting treatment from doctor.</td>
</tr>
<tr>
<td>3.</td>
<td>Spending a lot of time to understand the tools which use in this project</td>
<td>Medium</td>
<td>Medium</td>
<td>Reading different books and sources and Consult with supervisor to help prioritise tools</td>
</tr>
<tr>
<td>4.</td>
<td>Difficulties in fulfilling some requirements</td>
<td>Medium</td>
<td>Low</td>
<td>Getting help from the supervisor, using Internet forums where I can ask experts in JavaScript, and reading more books.</td>
</tr>
<tr>
<td>5.</td>
<td>Passing the deadline without completing the task</td>
<td>Low</td>
<td>Medium</td>
<td>Asking for extension.</td>
</tr>
<tr>
<td>6.</td>
<td>Hardware problem (Laptop)</td>
<td>Medium</td>
<td>Low</td>
<td>Try to fix it or use another one.</td>
</tr>
<tr>
<td>7.</td>
<td>The system is attacked by hackers or intruders</td>
<td>High</td>
<td>Low</td>
<td>Getting help from HWU IT people.</td>
</tr>
</tbody>
</table>

Table 4: Risk Plan Table
4.3 Conclusion

In conclusion, the main purpose of this project is demonstration of A Web-based GPS Data Visualisation tool for Sport Scientists and allows the new business can gain benefits from this software. This project will probably take 85 days approximately in order to develop and deliver the software within expected time. In addition, it should meet all of the requirements and accomplish the goal of this project. However, this project should follow the ethics which are defined in this report. Finally, the most challenge of development life cycle of this project is how to apply the various web development tools for making this web application to be a successful project.
Chapter 5

Design and Implementation

5.1 Overview of design and implementation

The design and implementation phase described how the system structure was designed and its interaction with the end user (director of HPSI Ltd., and sport scientists) and how the application is deployed in the field of interest. It details the application with regards to each component and the functions and the hardware requirement that will enable the application to run efficiently. A use case diagram is used to show an explicit interaction between the user and the system and a data flow diagram represents the processing of input and output data. Each piece of code that implements an important function is explained with screen shots for proper understanding. In addition, user manual of this application can be found in appendix B.
5.2 System structure chart

The purpose of a structure chart is to provide a basic graphical representation of the complicated processes within the system. This project comprises several modules including user registration, the uploading of multiple GPS data files, pitch and players visualisation, insertion of a number of rows and columns and the speed/distance of each player’s visualisation.
A detailed description of each subsystem is shown below:

5.2.1 User registration
This subsystem will allow the user to register before accessing an application.

Input personal data - To access the system, the user must input first name, last name, username, password, and email address to be identified by the system.

Verify approval - This subsystem allows only verified users to access the system.

Display valid account - This subsystem returns the account to the user setting.

5.2.2 Uploading multiple GPS data files
This subsystem will allow the user to upload multiple GPS data files before visualising the analysed data.

Input multiple GPS data files – To identify the file directory from the user.

Collect the file into system – To store the GPS data files in the system.

5.2.3 Pitch and players’ visualisation
This subsystem visualises the pitch and players on screen.

Get time, longitude, and latitude of each player Reads the values of time, longitude, and latitude from the input file.

Process the location of players and apply to the pitch – Convers the location from the file into the visualised pitch.

Visualise the pitch and how players move – Animates players’ movement on the visualised pitch.

Count the frequency of football player passes in each area – Records time spent by each player in particular areas.

Represent the frequency of football player passes in each area in colour code - Colour codes the visualised pitch according to time spent in each area.
5.2.4 Insertion of number of rows and columns

This subsystem will allow the user to input a number of rows and columns to visualise the pitch.

- **Input number of rows and columns** – To collect the number of rows and columns from the user.

- **Analyse input data** – To analyse and apply a new number of rows and columns.

- **Update and visualise new number of rows and columns** – To visualise the pitch according to the new number of rows and columns.

5.2.5 Speed/distance of each player visualisation

This subsystem will generate the graph from the data analysis process.

- **Get time, longitude, and latitude of each player** – To read the values of time, longitude, and latitude from the input file.

- **Perform and analyse data** – To calculate the speed and distance covered for each player from GPS data.

- **Display the speed/time and distance/time visualisation in graphic terms** – To visualise the speed and distance graph on the screen.
5.3 Use case Diagram

5.4 Context data flow diagram
5.5 Hardware and System Environment

- Operating System and Utilities Applications
  - Microsoft Window 8
  - Google Chrome version
  - Firefox
- Web Server Software
  - Appserv 2.5.10
- Editor
  - Editplus
- Database Management System (DBMS)
  - SQL server is a database management is used for store data
- Programming and Scripting Tools
  - PHP
  - HTML
  - JavaScript - jQuery
- Components
  - GoogleAPI is used for generating the graph

As explained in the literature review in Chapter 2 this application will use processing.js to visualise data on web-based application. In practice, processing.js has a number of limitations. The syntax of processing.js is similar to that of the Java language, so combining the Java language and web development languages in the same application is confusing and may prove frustrating to read and edit for other future developers. In addition, processing.js generally requires .pde files and can only access normal JavaScript code on the same page. It lacks direct communication between different canvases. jQuery, on the other hand, has a fast and concise JavaScript Library which simplifies HTML document traversing, enabling animation as well as work with multiple files for rapid web development. There are eight good reasons to support the use of jQuery (Chen, 2010) to develop the web application. Firstly, jQuery can work cross-browser and is able to run as the major browser. Its core library is small so it can be easily and rapidly included in the application. It has great documentation which enables developers to become familiar with it within a short time. It allows for the easy creation of several plugins. It fully supports the CSS3 selector specification and it encourages the building of utility functions that implement common functions usefully. It supports higher-
level constructs such as jQuery User Interface (jQuery UI), sliders, dialog boxes, date pickers etc. Finally, it is Widespread Adoption and a vast number of developers use it. Given the reasons above the researcher concluded that using jQuery would be of more benefit to the project than using processing.js.

5.6 Upload multiple GPS data files

To allow sport scientists or any other users to upload multiple GPS data files relating to different players at the same time, some important JavaScript Functions had to be used in this application. These were:

- `$('#fileUploadForm')`
  Function for notifying which function should be executed; ShowRequest(), AjaxError(), SubmitSuccessful().
- `ShowRequest()`
  Function for optional data processing before submitting the form.
- `AjaxError()`
  Function for letting users know there are some problems with data.
- `SubmitSuccessful()`
  Function for execution after files are uploaded successfully.
A Web-based GPS Data Visualisation tool for Sport Scientists

Figure 8: This figure represents the line of code in home.php to be processed after the user uploads GPS files

```
169  $(document).ready(function() {
170     $('#FileUploadedForm').ajaxForm({
171         beforeSubmit: ShowRequest,
172         success: SubmitSuccessful,
173         error: AjaxError
174     });
175     function ShowRequest(formData, jqForm, options) {
176         var queryString = $.param(formData);
177         alert('BeforeSend method: \n\nAbout to submit: \n\n' + queryString);
178         return true;
179     }
180     function AjaxError( data ) {
181         alert(data);
182         alert("An AJAX error occurred.");
183     }
184     function SubmitSuccessful(responseText, statusText) {
185         try {
186             file = JSON.parse(responseText);
187         } catch(e) {
188             alert(responseText);
189         }
190         var array = [];
191         var max = file[0].vel_array.length;
192         var player0LongestTime = 0;
193         for (var i = 1; i < file.length; i++) {
194             if (max < file[i].vel_array.length) {
195                 max = file[i].vel_array.length;
196                 player0LongestTime = i;
197             }
198         }
199         for (var i = 0; i < max; i++) {
200             var tempArr = [];
201             for (var j = 0; j < file.length; j++) {
202                 if (j == i) {
203                     tempArr.push(file[player0LongestTime].vel_array[i].time.toString());
204                 }
205                 if (i in file[j].vel_array) {
206                     tempArr.push(file[j].vel_array[i].vel);
207                 } else {
208                     tempArr.push(0);
209                 }
210             }
211             array.push(tempArr);
212         }
```

After users submit the GPS data files (Figure 8), the system counts the number of input files and provides a new file path for those files. Then, the `move_uploaded_file()` function moves the uploaded file to a new location (Figure 9).

```
10  for($i=0; $i<count($_FILES['fileToUpload']['name']); $i++) {
11      &vel_array = array();
12      $minute = 0;
13      $sec = 0;
14      $hour = 0;
15      $length = 0;
16      //Get the temp file path
17      $tmpFilePath = $_FILES['fileToUpload']['tmp_name'][$i];
18      //Make sure we have a filepath
19      if ($tmpFilePath != ""){
20          //Setup our new file path
21          $newFilePath = "/upload/" . $_FILES['fileToUpload']['name'][$i];
22          //Upload the file into the temp dir
23          if(move_uploaded_file($tmpFilePath, $newFilePath)) {
24```

Figure 9: The example code to shows transfer from user directory to new location
In the file upload.php (Figure 10), the data in CSV format will be read by function fopen(); this will parse each line in the file and return an array containing the fields read. In addition, the first line which has time, latitude, and longitude data will start on line 8 and should be in column A(0) = time, F(5) = latitude, and G(6) = longitude in CSV file (Figure 11). The data will then be stored in an array. In terms of preparing data for visualisation of speed and distance for each player (Line Chart), this file will calculate the distance using the formula:

\[ d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \]

And speed using the \( V = \frac{S}{T} \) formula, returning speed per 60 seconds / minute. Figure 12 shows an example of the interface which will allow the user to upload the GPS data files.
A Web-based GPS Data Visualisation tool for Sport Scientists

Figure 11: The example of .csv file

Figure 12: Interface of uploading file function
5.7 The pitch and each player can be visualised on the field

The raw coordinates of the pitch or latitude and longitude in the real world cannot be drawn onto canvas or visualised directly in the application, so the coordinates in the user’s multiple GPS data input files need to be converted (Figure 16). Firstly, JavaScript code in Figure 14 will get the value (canvas) from the HTML code (Figure 13).

```
var c = document.getElementById("canvas");
var ctx = c.getContext("2d");
//calculate from coordinate of top left
//from this site https://www.opengis.net/degree/llavol.html
var lengthOfADegreeOfLatitude = 111311;
var lengthOfADegreeOfLongitude = 84961;
var rotatedCoordinates = rotateCoordinates();
var leftTop = rotatedCoordinates[0][0];
var widthTop = Math.abs(rotatedCoordinates[0][0] - rotatedCoordinates[0][1]) * lengthOfADegreeOfLatitude;
var widthBottom = Math.abs(rotatedCoordinates[0][1] - rotatedCoordinates[0][2]) * lengthOfADegreeOfLatitude;
width += 10;
if (widthTop > widthBottom) {
    latWidth = widthTop;
} else {
    latWidth = widthBottom;
}
lonTop = rotatedCoordinates[0][1];
var heightLeft = Math.abs(rotatedCoordinates[0][1] - rotatedCoordinates[1][1]) * lengthOfADegreeOfLongitude;
var heightRight = Math.abs(rotatedCoordinates[0][1] - rotatedCoordinates[2][1]) * lengthOfADegreeOfLongitude;
height += 5;
if (heightLeft > heightRight) {
    lonHeight += heightLeft;
} else {
    lonHeight += heightRight;
}
```

Figure 13: HTML code to call canvas

Figure 14: JavaScript code to draw visualised pitch on the web application from actual data

Since the actual coordination of width of the top and bottom of a pitch are different in order to get the width of the pitch drawn on an application the larger width should be used in the visualised pitch. The system will automatically add 10 in to the width of the top if the width of the top is larger than the width of the bottom, and 10 in to the width of the bottom if the width of the bottom is larger than the top. The height of the pitch is worked out using the same logic but 5 in are added instead of 10 in.
A Web-based GPS Data Visualisation tool for Sport Scientists

The coordinates of the pitch at the Heriot Watt University were used as default coordinates for each corner of the visualised pitch, as shown in Figure 15.

Figure 15: The actual coordination from the pitch in Heriot Watt University

```
function rotateCoordinates() {
    //set default value for coordination of each corner
    var topleftX = 55.939466;
    var topleftY = -3.232073;
    var bottomLeftX = 55.939374;
    var bottomLeftY = -3.233092;
    var topRightX = 55.938589;
    var topRightY = -3.231760;
    var bottomRightX = 55.938477;
    var bottomRightY = -3.232771;

    var w = (bottomLeftX - topleftX)*180/PI*DegreeOfLatitude;
    var h = (bottomLeftY - topleftY)*180/PI*DegreeOfLongitude;
    var rotation = Math.atan(w/h) * (180/Math.PI);
    if (r>0) {
        rotation+=90;
    }
    var xm = topleftX;
    var ym = topleftY;
    var topleft = [topleftX, topleftY];
    var bottomleft = rotate(bottomleftX, bottomleftY, xm, ym, rotation);
    var topright = rotate(topleftX, topleftY, xm, ym, rotation);
    var bottomRight = rotate(bottomRightX, bottomRightY, xm, ym, rotation);
    return {topleft, bottomleft, topright, bottomRight, rotation};
}
```

Figure 16: The lines of code to show rotateCoordinates() function
The function `rotateCoordinates()` (Figure 16) will identify the coordinates of each corner from their longitude and latitude, then this function will convert these coordinates and draw the visualised pitch using the `rotate()` function (Figure 17).

```
135 function rotate(x, y, xM, yM, a) {  
136    var cos = Math.cos,
137        sin = Math.sin,
138        a = a * Math.PI / 180, // Convert to radians because that's what
139        // JavaScript likes
140        // Subtract midpoints, so that midpoint is translated to origin
141        // and add it in the end again
142        xR = (x - xM) * cos(a) - (y - yM) * sin(a) + xM,
143        yR = (x - xM) * sin(a) + (y - yM) * cos(a) + yM;
144    return [xR, yR];
145 }
```

Figure 17: The rotate() function
5.8 Visualise player and animate their movement using the circle, with a specific number according to the order in the input file

When sports scientists analyse sports matches, they usually refer to different zones of the pitch, and events that happen in those specific zones. They are also usually interested in the times that different players spend in those zones. Therefore, the basic visualization in our system is based on zones (which will be configurable by the user). The zones will be colour coded to show the relative amount of time that a specific player (chosen by the user) has spent in that zone throughout the time recorded in the GPS data file. In addition, each player is represented by a circle and within each circle is a specific number identifying each player, ordered according to the priority of the uploading file (Figure 18)

![Figure 18: The example interface of players' visualisation on an application](image)

In the `drawPlayer()` function (Figure 19), a loop rotates the player's coordination in the file placing the new `lat` variable and `lon` variable into `playerCoordinates` variable according to the origin (coordination of top left) and rotation (Tangent Function) of the pitch until the file has no variable in terms of coordinate.
5.9 Provide a table to represent number for players associated with their name

The system provides the specific number of players with their names on the left of the screen (Figure 20).
In the `tableCreate()` function (Figure 21), the system obtains the value of the number of input files (i+1, it plus 1 because the number of input files starts from 0) and the name of each player depending on the ordered input file (file[i].name).

```javascript
function tableCreate(file)
//create table for identify the number and the name for each player
var divTable=document.getElementById('divTable');
var tbl=document.createElement('table');
tbl.style.width='100%';
tbl.setAttribute('border','1');
var thead=document.createElement('thead');
var tr=document.createElement('tr');
var th=document.createElement('th');
th.appendChild(document.createTextNode("No."))
tr.appendChild(th);
var th=document.createElement('th');
th.appendChild(document.createTextNode("Name"))
tr.appendChild(th);
thead.appendChild(tr);
var tbody=document.createElement('tbody');
for(var i=0;i<file.length;i++)
    var tr=document.createElement('tr');
    var td=document.createElement('td');
    td.appendChild(document.createTextNode(i+1))
    tr.appendChild(td);
    var td=document.createElement('td');
    td.appendChild(document.createTextNode(file[i].name))
    tr.appendChild(td);
    tbody.appendChild(tr);
tbl.appendChild(thead);
tbl.appendChild(tbody);
divTable.appendChild(tbl)
}
```

Figure 21: The code represents how to extract the number of players and their names
5.10 Allow user to input the number of rows and columns

Generally, the visualised pitch will have a default number of rows and columns, as in Figure 23 (row = 12 and column = 6). After the user uploads the files, the system will allow the user to input the required number of rows and columns (Figure 24). It provides a simple 'analysis' button which, when pressed, divides the field on the basis of user input.

```
row = 12;
column = 6;
grids = calculateColourCodedGrids();

function drawLines(arow) {
    var widthScale = c.width/column;
    var heightScale = c.height/row

    for (var i = 1; i < column; i++) {
        ctx.moveTo(i*widthScale,0);
        ctx.lineTo(i*widthScale,c.height);
        ctx.stroke();
    }

    for (var i = 1; i < row; i++) {
        ctx.moveTo(0,i*heightScale);
        ctx.lineTo(c.width,i*heightScale);
        ctx.stroke();
    }
}
```

**Figure 22:** The 355\(^{th}\) and 356\(^{th}\) line in the code to set the default value of number of rows and columns

**Figure 23:** Interface of an application for user to input number of row and column
function changeRowColumn() {
    var temp_row = parseInt(document.getElementById('inpRow').value);
    var temp_column = parseInt(document.getElementById('inpColumn').value);

    if (!isNaN(temp_row) && !isNaN(temp_column)) {
        if (temp_row % 1 !== 0 || temp_column % 1 !== 0) {
            row = temp_row;
            column = temp_column;
            grids = calculateColourCodedGrids();
        }
    }
}

5.11 The grid is colour coded based on the time that the players spend in each part.

Figure 24: changeRowColumn() function will execute when user inputs a new number of row and column.

Figure 25: An interface of time spent of each player in particular area.
In the visualised pitch, there are colour codes in the visualised pitch (green, yellow, and red) according to time spent in each particular area, for each player (Figure 25). First, the `calculateColourCodeGrid()` function in Figure 26 will calculate and retrieve the actual player's coordinates; then it will rotate these coordinates to match with the correct coordinates on the canvas (visualised pitch). The system will then calculate which grid the user is in and that grid will add 1 for each row of data. As a result the data is stored in a 2-d array.

```javascript
function calculateColourCodeGrid() {
  var e = document.getElementById("selectPlayerForGrid");
  var $ul = $(e).children();
  var player = -1;
  var stp = 0;
  var w = document.getElementById("canvas");
  var widthScale = w.width/column;
  var heightScale = w.height/row;
  var startRow = 500;
  for (var j = startRow; j < file[player].array.length; j++) {
    var playerCoordinates = rotate(file[player].array[j].lat, file[player].array[j].lon, rotatedCoordinates[0][0], rotatedCoordinates[0][1], rotatedCoordinates[1][0], rotatedCoordinates[1][1]);
    // rotate the player's coordinates from the file and put the new lat lon into playerCoordinates variable
    // rotatedCoordinates[0][0] and rotatedCoordinates[0][1] are the origin of the rotation (top-left)
    // rotatedCoordinates[1][0] and rotatedCoordinates[1][1] are the origin of the rotation (top-left)
    var x = Math.abs(playerCoordinates[1]-wLeft)*lengthDegreeToPixel/width;
    var y = Math.abs(playerCoordinates[0]-wTop)*lengthDegreeToPixel/width;
    var gridX = Math.floor(x/widthScale);
    var gridY = Math.floor(y/heightScale);
    if (!gridX.toString() in obj) { obj[gridX.toString()] = [];
      obj[gridX.toString()][gridY.toString()] = 1;
    } else { obj[gridX.toString()][gridY.toString()] += 1;
    }
  }
  return obj;
}
```

Figure 26: `calculateColourCodeGrid()` function

```javascript
function drawColourCodeGrid(xRow) {
  var widthScale = c.width/column;
  var heightScale = c.height/row;
  var totalGrids = column*row;
  for (var i = 0; i < column; i++) {
    for (var j = 0; j < row; j++) {
      if (i.toString() in grid[i.toString()]) {
        var counter = 0;
        for (var l = 0; l < column; l++) {
          for (var m = 0; m < row; m++) {
            if (m & p & parseInt(grid[l.toString()][m.toString()])) {
              if (grid[l.toString()][m.toString()] > grid[i.toString()][m.toString()]) {
                counter++;
              }
            }
          }
        }
        if (counter >= totalGrids*3/10) {
          ctx.fillStyle = "red";
        } else if (counter >= totalGrids*2/10) {
          ctx.fillStyle = "yellow";
        } else if (counter >= totalGrids*1/10) {
          ctx.fillStyle = "green";
        } else {
          ctx.fillStyle = "white";
        }
      }
    }
  }
}
```

Figure 27: `drawColourCodeGrid()` function to colour the pitch
In the `drawColourCodeGrid()` function (Figure 27), there is a counter variable which is used to calculate the percentage of latitude and longitude for each row. The counter is increased by 1. For example, if the counter variable of a particular area has 90% more player’s movement than any other grid it will be coded ‘red’; 60% will be coded ‘yellow’, and 20% will be coded ‘green’.

5.12 A drop down box for player selection to colour the pitch

![An interface of drop box for player selection](image)

The system will provide a dropdown function (Figure 29) to allow the user to pick the colour for the player, to be placed on the visualised pitch according to time spent in each area.
A Web-based GPS Data Visualisation tool for Sport Scientists

5.13 Visualisation of a speed/time and distance/time graph for each player

According to the upload.php file, the data is set in the well form (Figure 10) and this data is used in the drawChart() and drawChart2() function through the Google API chart (Figure 30 and Figure 31).

```javascript
function createDropdown() {
  var dropdown = document.getElementById("selectPlayerForGrid");
  for (var i=0; i<file.length; i++) {
    var option = document.createElement("option");
    option.text = (i+1).toString();
    option.value = i;
    dropdown.add(option);
  }
  document.getElementById("selectPlayerForGrid").style.display = "block";
}
```

Figure 29: createDropdown() function

```javascript
// draws the chart
function drawChart() {
  document.getElementById("chart_div").style.display = "block";
  // Create the data table,
  var data = new google.visualization.DataTable();
  data.addColumn("string", 'Time');
  for (var i = 0; i < file.length; i++) {
    data.addColumn("number", file[i].name);
  }
  data.addRows(arrays);
  // Set chart options
  var options = {'title': 'The speed (meters)',
                 'hAxis': {'title': 'time (minutes)'};
  // Instantiate and draw our chart, passing in some options.
  var chart = new google.visualization.ColumnChart(document.getElementById("chart_div"));
  chart.draw(data, options);
}
```

Figure 30: The function drawChart() is used to visualise the speed and time graph.

57
function drawChart2() {
    document.getElementById("chart2_div").style.display = "block";
    // Create the data table.
    var data = new google.visualization.DataTable();
    data.addColumn('string', 'Time');
    for (var i = 0; i < file.length; i++) {
        data.addColumn('number', file[i].name);
    }
    data.addRows(array2);
    // Set chart options
    var options = {title:'The distance (meters)',
                    hAxis: {title: 'time (minute)'}
                };
    // Instantiate and draw our chart, passing in some options.
    var chart = new google.visualization.LineChart(document.getElementById('chart2_div'));
    chart.draw(data, options);
}

drawChart2();

Figure 31: The function drawChart2() is used to visualise the speed and time graph

Figure 32: An interface of speed over time graph
5.14 The security system

Security is an important issue which needs to be addressed in this study. The players use the system to work out their general level of fitness and their data is important to them as it will help them assess their future performance. Therefore, the system needs to be secure and needs to have checks which ensure that the player has consented to their GPS data being viewed by the sports scientist. The application provides a login system with a high verification level of registration. In terms of authorised users, there are two levels of users. The first level of users are those who have the status of ‘user’ (it will specify such when users register on the system), and they can access and take advantage of any of the features of the application, but they cannot create new users. Second level are users those who have the status of ‘administrator’. They can use all the functions of the application, but they can also create new users. Therefore, the registration function (Figure 32) will require an existing authorised user with ‘administrator’ status to approve a new user in order to prevent any users registering onto the system.
A Web-based GPS Data Visualisation tool for Sport Scientists

Figure 34: An interface of registration page

Figure 35: An example code to ensure new user input the required information
5.15 Summary

In the software development process the design phase is an important phase which can determine the success of a project and allow the developer to see the possible functions of an application. Diagrams were used to help illustrate and present a clear picture of the system. Structure charts, use case diagrams, and context data flow diagrams are used extensively in this project. The structure chart provides a clear explanation of each subsystem of the application. Before introduction how to implement this application, hardware and system environment during software development process were identified. Moreover, there is a list of the main functions of this application include:

- Upload multiple GPS data files
- The pitch and each player can be visualised on the field
- Visualise player and animate their movement using the circle, with a specific number according to the order in the input file
- Provide a table to represent number for players associated with their name
- Allow user to input the number of rows and columns
- The grid is colour coded based on the time that the players spend in each part
- A drop down box for player selection to colour the pitch
- Visualisation of a speed/time and distance/time graph for each player
- The security system

Finally, an explanation of each functionality feature with a piece of code and an example of interfaces were also provided to allow other developers to understand exactly how to implement this application.
Chapter 6

Testing and Evaluation

6.1 Overview of testing and evaluation

This chapter describes the means of testing and evaluation of the entire application to establish the system response ability, and to achieve the requested process efficiently. Any issues encountered should be resolved after testing and evaluation to improve the performance of the application and gain highest user satisfaction. This project involves a number of specific software components: secure login, GPS file-upload (single and multiple), GPS-file parsing, and a number of specific visualisation. Each of these needed to be developed and tested independently, to ensure correct functionality, using (where relevant) example GPS files provided by the NPCS. In addition, the complete system also needed to be evaluated by at least one potential user (a sports scientist) to provide constructive comments for further development. This chapter describes first the tests undertaken to ensure correct functionality, and second the evaluation of the complete system by a potential user.

6.2 Testing

The three testing strategies that would be adopted are component, integration and usability testing to identify critical bug/errors, fix them and thus make sure that this application meets all user requirements.

6.2.1 Component Testing

To ensure that the application can produce the correct response and result according to the user’s request, a piece of the code is tested by unit testing. It will be tested several times for both functional and non-functional requirements before delivery to the users and deployment in the real system.
## 6.2.1.1 User registration and log in system

<table>
<thead>
<tr>
<th>Operation Performed</th>
<th>Condition Tested</th>
<th>Actual Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill in the name and last name of user</td>
<td>The name and last name textbox should allow user to input information.</td>
<td>Pass</td>
</tr>
<tr>
<td>Fill in new unique username and password</td>
<td>The username textbox should allow user to input and password should be invisible.</td>
<td>Pass</td>
</tr>
<tr>
<td>Fill an email and identify log in status</td>
<td>An email address textbox should allow user to input information and be able to select appropriate status.</td>
<td>Pass</td>
</tr>
<tr>
<td>Fill authorized username and password for approval</td>
<td>The authorized user should have an administrator privilege.</td>
<td>Pass</td>
</tr>
<tr>
<td>Incomplete entry of required user information e.g. name or password</td>
<td>The system must inform to the user to fill in the blank.</td>
<td>Pass</td>
</tr>
<tr>
<td>Fill invalid authorized username and password for approval</td>
<td>The system should not allow user to complete registration process.</td>
<td>Pass</td>
</tr>
<tr>
<td>Input with valid username and password to log in</td>
<td>The system should allow the user to access to the system.</td>
<td>Pass</td>
</tr>
<tr>
<td>Input with invalid username and password to log in</td>
<td>The system should not allow the user to access to the system.</td>
<td>Pass</td>
</tr>
</tbody>
</table>
### 6.2.1.2 Uploading multiple GPS data files

<table>
<thead>
<tr>
<th>Operation Performed</th>
<th>Condition Tested</th>
<th>Actual Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload single GPS data file with the correct file format</td>
<td>The file input form should allow user to be able to upload the file.</td>
<td>Pass</td>
</tr>
<tr>
<td>Upload multiple GPS data files (up to 9 files) with the correct file format</td>
<td>The file input form should allow user to be able to upload the files.</td>
<td>Pass</td>
</tr>
<tr>
<td>Upload single GPS data file with the incorrect file format</td>
<td>The file input form should not allow user to be able to upload the file.</td>
<td>Pass</td>
</tr>
<tr>
<td>Upload multiple GPS data files (up to 9 files) with the incorrect file format</td>
<td>The file input form should not allow user to be able to upload the files.</td>
<td>Pass</td>
</tr>
</tbody>
</table>

### 6.2.2.4 Speed and distance of each player visualisation

<table>
<thead>
<tr>
<th>Operation Performed</th>
<th>Condition Tested</th>
<th>Actual Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visualise the speed of each player from single GPS data file</td>
<td>The graph is visualised within 5 seconds. With graphical representation of player associated with a colour code for his speed.</td>
<td>Pass</td>
</tr>
<tr>
<td>Visualise the speed of each player from multiple GPS data file</td>
<td>The graph is visualised with graphical representation of player associated with a colour code for their speed.</td>
<td>Pass</td>
</tr>
<tr>
<td>Visualise the distance of each player from single GPS data file</td>
<td>The graph is visualised within 5 seconds. With graphical representation of player associated with a colour code for his speed.</td>
<td>Pass</td>
</tr>
<tr>
<td>Visualise the distance of each player from multiple GPS data file</td>
<td>The graph is visualised with graphical representation of player associated with a colour code for their speed.</td>
<td>Pass</td>
</tr>
</tbody>
</table>
6.2.2.3 Pitch and players visualisations

<table>
<thead>
<tr>
<th>Operation Performed</th>
<th>Condition Tested</th>
<th>Actual Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visualise the pitch and player from single GPS data file</td>
<td>The pitch and player is visualised within 5 seconds.</td>
<td>Pass</td>
</tr>
<tr>
<td>Visualise the pitch and player from multiple GPS data files</td>
<td>The pitch and player is visualised.</td>
<td>Pass</td>
</tr>
<tr>
<td>Select the player to colour the pitch based on his time spent</td>
<td>The pitch texture represents how often the selected player moves around a particular circumference of the pitch.</td>
<td>Pass</td>
</tr>
<tr>
<td>Input number of rows and columns to visualise in the pitch</td>
<td>The pitch should be change according to user input within 5 seconds.</td>
<td>Pass</td>
</tr>
</tbody>
</table>

6.2.2 Integration Testing

The purpose of integration testing is to ensure that the system can operate correctly according to the specifications. These activities are carried out after a successful integration of all the functions of the application. In terms of testing there are two phases which are considered in order to test all mandatory functional aspects of the application.

6.2.2.1 Registration and log in system

At this point, after the user has registered and been identified and approved by the administrator, the system should allow the user to proceed to login to the application with a valid username and password in order to access the main work space of the application.

6.2.2.2 Uploading GPS files and analysed data visualisation

In the work space area, the user browses to locate the single or multiple GPS data files with appropriate file format, then uploads the files. The system will take a few seconds, depending on the number of files, before it visualises the pitch, the players with their
name, the colour code of each grid, and speed/time and distance/time graph on the browser. In addition, the user can input the number of rows and columns, then the visualised pitch should be changed according to the user’s input when the ‘analyse’ button is pressed.

### 6.2.3 Usability Testing

Usability testing is done to evaluate the usability of the application; during the software development life cycle the sport scientist, athlete and other stakeholders will use the application to perform specific tasks several times in order to learn how to use the application. In terms of user satisfaction, all users will evaluate and identify which functions should be improved to enhance the performance and effectiveness of the system.

### 6.3 Evaluation

After the researcher has tested the complete system several times in order to ensure that it meets all user requirements to the user’s satisfaction, the application should be evaluated by sport scientists at the Heriot Watt University. The criteria and purpose of evaluation should be to confirm that all the mandatory requirements have been met. The feedback from the evaluation is then gathered and any features which have been missed will be added. The evaluation methodology which was applied in this project was an online questionnaire comprising concise but comprehensive questions which covered all aspects according to the objectives of the evaluation. After gathering the feedback and suggestions some additional features were added and others were earmarked for future development. Moreover, the questionnaire results helped to indicate the strengths and weaknesses of the application.
6.3.1 Questionnaire results

The researcher met with the sport scientist at the Heriot Watt University, Dr. David Sykes, Sports Science Manager for Heriot-Watt University and former Lead First Team Sports Scientist for Hearts FC and demonstrated the application, after which the sport scientist was asked to fill out the online questionnaire to evaluate the system. The sport scientist’s evaluation showed that he strongly agreed that this application meets all functional requirements in the performance and security issues as list below:

- The system can upload multiple GPS data files.
- The pitch and each player can be visualised on the field.
- Each player is visualised by a circle with a specific number according to the order of input file.
- A table is provided to represent the number of the players associated with their name.
- The system can animate all of the players (a player is shown as a circle on the screen) and automatically show players’ moves.
- The system allows users to input the number of rows and columns by providing a simple 'analysis' button; when pressed, the field is divided on the basis of the user’s input.
- The grid is colour coded according to the length of time the player has spent within the grid.
- The system provides a drop down box to select the player and his field is colour-coded based on his moves.
- The system can visualise a speed/time graph for each player.
- The system Visualise a distance/time graph for each player.
- The security measures are appropriate to this application.

In terms of Graphical User Interface, the sport scientist agreed that the layout was clear and understandable, the design was user-friendly, and the colouring was appropriate. However, operation or function of an application meets user-experience and easy to use gain agrees neutral. Therefore, although this application meets the user requirements and user-friendly criteria for the sport scientist, it might be difficult to develop this application to make it user-friendly for other users.

There were three open-ended questions in the questionnaire to allow participants to give their own feedback and some suggestions to improve this application. The first suggestion was that there should be an indication of what each colour on the pitch relates to. Secondly, the system should allow the user to view the graph over a specific period (future work). For example, to allow the analysis of average speed over five minutes. Other general comments on
how to turn this application into a more useful tool for practitioners included: integrating a function to enable the addition of time splits to allow data to be removed when a player is substituted and/or break down data into different halves of the game; making it possible to enter co-ordinates for different pitches and to distinguish between players on different teams (future work). All suggestions made can be found in Appendix A.

6.4 Conclusion

After using the three strategies to test the entire system in both functional and non-functional requirements to ensure that the application is of high quality it resulted that the application shows the ability to be accessed any time at the user’s request and provide a reliable response. Therefore, the application meets the user requirements as proposed. Although this system was tested many times, it still required evaluation from real users or sport scientists in order to get further feedback and suggestions. Sport scientists from the Heriot Watt University agreed that this application meets the core functional and non-functional requirements. However, there are a few further functionality features which should be added to make this application more useful.
Chapter 7

Achievement and Limitation

7.1 Overview of Achievement and limitation

This chapter presents the achievements and outlines the limitations encountered in actualising the aim of the GPS visualisation tool for sport scientists. In addition, it will cover through the challenges and the future work of this project.

7.2 Achievements

This project will be deemed successful if it meets the aims and objectives of the project and sport scientists, athletes, coaches, software developers and others are able to gain benefit from the GPS data visualisation application. The level of success of the project is measured by the level of satisfaction with the application and the fact that it met all user requirements. Because of the various browsers currently used by different users, the application was designed and tested to perform efficiently on several browsers including Google Chrome, Mozilla Firefox, and Safari. In terms of interface, this application visualises the information in a rich GUI (Graphical User Interface) to allow users to interact with its entire functionality easily. According to the evaluation (questionnaire results) by a sport scientist, who is a non-computing expert, this application is easy to use, its layout is clear and understandable, its design is user-friendly, and the colour coding is appropriate.

Functionality is measured in terms of the system’s capability of uploading GPS files according to the specified format; ability to visualise pitch and each player on the field of play; ability to provide tabular representation of each player with associated name; ability to animate the player in circular pattern moving automatically on the field; ability to provide a user with an input text for a number of rows and columns with an analysis button to display the visualised pitch based on user input; ability to indicate time spent by each player at a particular
A Web-based GPS Data Visualisation tool for Sport Scientists

area through differentiating using a colour coded grid; ability to graphically present a report chart for speed/distance with respect to time and the integral security system ensuring that only authorised users with valid credentials gain access to the application. In the software development life cycle, the testing phase of this application focused on user satisfaction of optional requirements and the quality of the software which has availability, accessibility, reliability, usability, and high performance.

This project brought a few challenges. Some of the selected tools mentioned in the literature review did not work well for this application in practice, so the software used for this project was changed to be able to support all the functionalities of the application and ensure the highest performance and effectiveness. Further problems included bugs and technologies which do not support some essential features. For example, the latest version of AppServ (AppServ 2.6.0) does not support phpMyAdmin so AppServ 2.5.10 was used instead. In addition, the researcher is not an expert in jQuery, which is the main language used to create this project, which led to a steep learning curve.

In conclusion, in order to achieve the aims of this project, the researcher has to be more proficient in web programming skills (HTML, JavaScript, PHP, etc.), must acquire new knowledge and figure out how to solve some bugs/errors. This project is not only of benefit to the new company, sport scientists and software developers, but is also of benefit to the researcher who has furthered his knowledge and experience of programming.
7.3 Limitation and future work

Although this project was successfully delivered to the end users and obtained their satisfaction, the application still has some limitations. The main limitation is that an internet connection is required to perform and operate this application. Although this application is not browser specific it cannot be operated on Internet Explorer. Additional features suggested by the sport scientist include allowing the user to view the graph in a specific time period to, for example, analyse the average of speed over five minutes. Moreover, the system should be able to add time splits to allow data to be removed when a player is substituted and/or break down data into different halves of the game. In addition, the ability to enter co-ordinates for different pitches and distinguish between players on different teams is important. Such additions would make this a more helpful tool for practitioners.
Chapter 8

Conclusion

Nowadays there are only a few available tools which enable the visualisation of data for sports scientists or sport clubs. Sports scientists generally use simple Excel functions to work out how strenuous the match or exercise was for each athlete and these functions cannot support visualisation of GPS data in a way that is helpful to sports coaches and athletes. Some cheap alternatives are typically not sports-specific and difficult to use, while some sports analysis software that is currently used is expensive. This project developed a prototype or demonstration of a tool that visualises and displays sports GPS data for sport scientists with the aim of enabling a new company to gain investment to develop the tool commercially. The four main objectives of this project were:

i) To produce a demonstration of a tool that visualises and displays sports GPS data.

ii) To demonstrate the potential advantages of this application for both sport scientists and the initial business company.

iii) To develop a user-friendly system.

iv) To build a system which can be extended and modified by non-experts.

The possible tools to develop the application and achieve the objectives of this project were thoroughly investigated in the literature review giving due consideration to the most appropriate technology for visualising GPS data over the web. Unfortunately, in practice, the tool mentioned in the literature review would not have been the most appropriate choice for this application and other tools and technologies were chosen, namely Appserv 2.5.10 as the main hardware and system environment and PHP, HTML, JavaScript as the main programming languages used. This project has both functional and non-functional requirements, as identified by the sport scientist. Care was taken to ensure that all the requirements to meet user satisfaction were identified in the design and implementation phase. To ensure the successful development and delivery of high quality software to the user several tests were carried out using various testing strategies (component, integration, and usability testing). The result of the
testing emphasised this software’s availability, accessibility, reliability. It also made the researcher aware of the limitations of the project. Before delivery of the application to the end user, the system was evaluated in order to gain feedback and recommendations from the Heriot Watt University sport scientist. The evaluation results highlight the achievements and future work required to improve on the application to make it more useful and effective for practitioners.

This project is beneficial to all stakeholders. For example it can help sport scientists to see the capabilities of HTML5, JavaScript, etc. in this context. In addition, this application helps sport scientists test certain ideas about what to visualise, what graphs to show, etc. As a result, this project has attracted the interest of a new company, a consortium comprising HWU computer scientists, sports scientists, and ‘High Performance Sports Innovation Ltd’, which is a new company started by such a consortium in order to commercialise the product. Although this project has some challenges and problems, it has been of great benefit to the researcher who gained the opportunity to apply web programming skills to solve problems in the application and find the solution by acquiring new knowledge from various materials. Consequently, the researcher has gained more experience in the field and has improved her programming skills in the software development process.
Bibliography


Appendix

Appendix A: Questionnaire

Questionnaire: A Web-based GPS Data Visualisation tool for Sport Scientists Project

*Required

Instructor(s): *

Date: *

Please indicate your role: *

Performance and security *Functional Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Strongly agree</th>
<th>Agree Neutral</th>
<th>Disagree</th>
<th>Strongly disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>The system can upload multiple GPS data files.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The pitch and each player can be visualised on the field</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visualise player by the circle with specific number according to the order of input file.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### A Web-based GPS Data Visualisation tool for Sport Scientists

<table>
<thead>
<tr>
<th>Feature Description</th>
<th>Strongly agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provide table to represent number for players associated with their name.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to animate all of the players (a player is shown as a circle on the screen) and see players’ moving automatically</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allow user to input the number of row and column by providing a simple ‘analysis’ button; when pressed, the field is divided base on the user input.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The grid is colour coded based on the time that player spent in that part.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provide a drop down box to select the player, and his field is colour-coded based on that player.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visualise a speed/time graph for each player.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visualise a distance/time graph for each player.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The system security is appropriate with this application.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A Web-based GPS Data Visualisation tool for Sport Scientists

Graphical User Interface *

<table>
<thead>
<tr>
<th></th>
<th>Strongly agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>The layout is clear and understandable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operation or function meets user-experience and easy to use</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Designing is user-friendly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>colouring is appropriate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Available color code provided by the system includes green, yellow, and red. What is your opinion about it, and/or suggest preferred color code?

How useful is the graphical representation of the speed/time and distance/time graph or you can suggest more helpful graphical representation?
Can you give the general comment or suggestion to this application?

<table>
<thead>
<tr>
<th>To improve the performance of the whole system.</th>
</tr>
</thead>
</table>

| A Web-based GPS Data Visualisation tool for Sport Scientists |

79
Appendix B: User Guide

A user guide is provided for user in order to have an easy user of the web-based GPS visualisation tool for sport scientist. This application has four phase for user to interact with this application. The first phase is the login, if the users has an existing account created. The second phase is for the user to register an account, if not already registered. The third phase is the main work space for visualising animated object. Finally, the fourth phase is a graphical representation of speed against time and distance against time.

The steps below are a guide on how to use this application without hitches after loading the application URL on any preferred web browser except an Internet explorer.

**User login**

To login as an existing user, type in your username and password and click on the Sign in tab to proceed as shown on the screen shot in Figure User Guide 1.

![Figure User Guide 1: Log in page](image)

For non-existing users create a new user account by clicking on register as show in the red circle on screen shot in Figure User Guide 2.
Figure User Guide 2: Link of register page

**Note:** After clicking on register, you will be prompted with a page to register as new member.
Registration of new user

On the registration page provide Administrator or user details as shown on Figure User Guide 3.

![User registration form]

**Figure User Guide 3: User registration form**

**Note:** The “Approved by” section is used for an existing user who has status is ‘Administrator’ to approve for new user.

After register or create a new user successfully, the login page will be automatically comes up for the user to input the username and password to proceed to the main work space.
Main work space

When login process successfully, you will be presented with a main page as shown on Figure User Guide 4.

![Main Work Space](image)

**Figure User Guide 4: Work space of an application**

The left hand side of the image in grey colour above consists of the name of the user currently accessing the application, the current workspace, and chart report area for graphical representation, print option, and logout.

The workspace area consist of a browse tab to locate GPS files according to the specified file format, the upload button to upload a file to the server, the textbox to input the number of rows and column to analyse and finally the analyse button to carry out the analysis of the uploaded files.
8 Steps to use the application

1. Click on the **Browse** button as shown on the Figure User Guide 4 to locate the file directory.
2. Select files which has a .CSV file extension. Then, click **Upload** button to upload to the server. (The application will divide the pitch into 12 x 6 for the first time automatically.)
3. Input the number of **Rows** and **Columns** to be visualised on the pitch and click on the **Analyse** button.
4. The work space will be visualised as shown in Figure User Guide 5
5. Chart report area will be visualised as shown in Figure User Guide 6 and 7.
6. Select player to represent time spent on the pitch from **Dropdown** function.
7. Click on **Print** button
8. Click on **Logout** button after finish

---

**Figure User Guide 5:** The screen shows the pitch, player, and player’s detail
A Web-based GPS Data Visualisation tool for Sport Scientists

Figure User Guide 6: The speed/time graph

Figure User Guide 7: The distance/time graph