Athlete's training website

Final Report for CS39440 Major Project
Author: Stoyko Abrashev (sta9@aber.ac.uk)
Supervisor: Dr. Andy Starr (aos@aber.ac.uk)

23rd August 2013
Version 1.2 (Release)

This report is submitted as partial fulfilment of a BSc degree in
Computer Science G400

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, UK

Declaration of originality

In signing below, I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for plagiarism and other unfair practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the sections on unfair practice in the Students' Examinations Handbook and the relevant sections of the current Student Handbook of the Department of Computer Science.
- I understand and agree to abide by the University's regulations governing these issues.

Signature …….. (Stoyko Abrashev)

Date ……

Consent to share this work

In signing below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Signature …….. (Stoyko Abrashev)
Date ……

Acknowledgements

I would like to gratefully acknowledge the valuable assistance of Dr. Chuan Lu, who gave me much advice and useful instructions on how to develop my project in the best way possible.

I would also like to thank my marker, Prof. Dave Barnes, for his guidance and recommendations on the implementation of my work.

My sincere thanks are also extended to my SAW supervisor, Dr. Andy Starr, and also to Mr Neil Taylor and Ms Janet Hardy for their encouragement and assistance.

Finally, I am very grateful to my parents and my sister for their loving support throughout my study.

Abstract

The present paper describes the creating of a web application for triathlon athletes who need to train by themselves. The final version of the project is an easy to use, good looking web page with a lot of functionality. All members have access to the system all the times and athletes are notified for new club information regarding their tasks.

The most important which make this project worthwhile is the technology that has been used. Most of the functionalities are executed client-side, but some things are processed on the server. The training system is implemented using jQuery libraries, Ajax and basic Javascript have been used very efficiently and are well implemented in the web page as well as the HTML5 language. I used JSON for storing and exchanging session information. The PHP language has been used to execute all commands needed to insert or read some information from the MySQL database.

Contents

1. Background & Objectives

1.1. Background

This product is intended to be used by a real client. The client is a triathlon club called INTRtri which is based in North Ceredigion in Mid West Wales. INTRtri Triathlon Club provide weekly sessions for swimming, biking and running, training events and individual coaching. INTRtri also offer for its members and non-members weekend sessions, e.g. bike rides, open water swimming in the summer, turbo trainer sessions and strength and conditioning sessions. For training individual clients, coaches draw up personal periodised training plans for athletes depending on their abilities and goals. My product will assist both coaches and athletes in their everyday activities. Coaches will be able to train athletes who are away from the club and have to train by themselves on daily basis. They can look at and analyze the training of their clients without requiring them to send in files or return sheets. When logging into the system, both coaches and athletes will have the full information about their current sessions.

For biking workouts, indoor turbo trainers or rollers may be used. They are an ideal tool for concentrating on specific exercises to improve the cycling performance, as it is possible, for instance, to simulate the size and steepness of a hill. Besides, there are instruments on some trainers to measure the power output, speed, cadence, heart rate. Rollers are closer to the actual feeling of riding a bike, so they are especially good for sharpening athletes' skills.
Treadmill would perfectly serve the purpose of indoor running exercise. It is possible to adjust training to your specific goal, e.g. you can combine running and walking at first to warm up then you can vary the speed and adjust the treadmill for a hill or a hike workout, or even a mountain climb workout.

Content of training
Session: type, description, duration, distance, by who, timeadded, intervals (each interval – description, duration and level of intensity). Levels are from 1 to 10.
Sample running session content:
Type: Running
Description: Note time and heart rate
Duration: 40 minutes
Distance: 6km
Interval 1 – Warm up – 10 minutes at level 4
Interval 2 – Run while nose breathing – 30 minutes at level 6.
Sample cycling session content:
Type: Cycling
Description: Stay within your endurance zone
Duration: 40 minutes
Distance: 16 km
Interval 1 – Turbo, rollers or spin class – 40 minutes at level 7.
Sample swimming session content:
Type: Swimming
Description: Fitness session
Duration: 40 minutes
Distance: 16 km
Interval 1 – Easy swim drills – 20 minutes at level 4
Interval 2 – Aqua jog steady 20 minutes at level 6.

Sessions are defined by the coach for each athlete (or a group of athletes). The coach can save a specific session, even if he does not have athletes to train at a given moment. If a

new athlete joins the club, the admin will save his/her data in the database together with the coach assigned. The coach can change a cycling, running or swimming session and save it as a new session. The data is set by the coach, sessions (just cycling and running) are then held in real time by athletes. The coach can monitor the performance of each athlete seeing if he/she has completed a given session or not.

Existing products and websites of similar functionality in the same field have been reviewed to find out appropriate sport-related information as well as to study the structure, functionalities, styling, overall performance, good practices, methods and development tools needed to build a properly working and user-friendly website. Some products that could be mentioned are as follows:
TrainingPeaks (14), Triathlete (15), Triathlon (16), England athletics (17).

Training Peaks
It is a web based easy-to-use software which is available online and through iOS and Android applications. They have a basic free version, that doesn't let you plan ahead. The full version runs about $80.00 a year, so it's very affordable. The cost for a year of TP is about the cost of one month of training. TP has it all integrated - planning, results, analysis, food.
Two different editions are available – one for coaches and another for athletes that help them monitor, analyze and plan workouts and nutrition. An excellent feature is the extensive device compatibility. They can upload workouts directly from the device to TrainingPeaks.
Athletes are able to plan ahead training, moreover, they get daily reminders about their training. They can choose a plan that is best suited to their goals. They can use the virtual coach and annual training plan to set up and periodise training. And finally, they can choose their personal coach. Athletes are able to log meals, workouts and metrics and view the information.
Coaches are given access to metrics and charts. They can monitor and analyze the athletes' training from everywhere. Coaches can add their libraries to the store and set their own price.
A great tool for planning, monitoring and analysing. It is possible to calculate TSS for cycling workouts, rTSS for running and sTSS for swimming.
A few drawbacks may be found, e.g., user interface is ineffective and not easy to navigate. Besides, iPad and mobile applications are poorly designed and not fully functional.

Building such a product may turn out to be very good idea from the commercial point of view as well. Some of products available in the market deal mainly with organizational aspects, as they are intended primarily for organizers of sports events; others offer valuable information, courses, individual planning and tracking of training for athletes, and training techniques and analyzing methods for coaches.

1.2. Objectives

The main objective of the present project is creating a web site for training triathlon athletes who are away from the club. The website will represent an integrated system allowing coaches and athletes to communicate and keep track of the training schedule. Athletes will be able to check their sessions in real time and get full information about them. There will be several qualified coaches. A coach will be able to specify the details of a training session and each athlete should be able to select a particular session and

receive real time information about it. Coaches will be able to create new sessions and edit them and assign specific sessions to individual or all athletes.

It is hoped that this application made in pursuing a Bachelor's degree in Computer Science will serve not only for academic purposes but will have some practical use as well. Moreover, the present version could be tailored to specific customers' needs to include not only triathlon, but other individual or team sports as well.

2. Requirements

2.1. Gathering requirements

The web site will provide an admin panel for each coach with all athletes trained by him and all the sessions made by him. The athletes should use the web site to know everything about every new session which they have to do. This system will be used by all current and new athletes and coaches. Some administrative functions must be done by administrator who has to be familiar with the management of the system.
About the Google map (12) at first I thought it will be a good idea to have a way of seeing where is the athlete in his journey (if he rides a bike or running). But after talking with my supervisor I understood that athletes will be training inside on walking path or simulated bike not outside. So the whole idea of updating athlete's location and sending it to database was not suitable for that kind of sport.

2.2. Functional requirements

- FR01 – Users. There are three types of users (coaches, athletes and an administrator). The following details are required for all of them – username, password, first name, last name, age, coach (0 or 1), trained_by (nobody for coach), email and picture.

a) Coaches – they have to login to add new sessions or edit a session.
b) Athletes – they have to login to check their new sessions.
Both coaches and athletes can change their username, password, email or picture.

c) Administrator – he is setup already in the database with id 1, his username is **administrator** and password is **thisclubadmin365**. Table structure and initial information about him is in **clubfinal.sql** which has to be imported in database with name "club". Then the Administrator has to register any new coach or athlete who joins the club. He also can delete any coach or athlete who leaves, change an athlete's coach as well as change any member's first name, last name and age. At last he is able to edit his own first name, last name, email or age.

- FR02 - Sessions. If you are coach it is possible to add new sessions and edit existing sessions. Athletes can view their new sessions.
- FR03 - Adding and editing a session. A session will consist of description, type of the session (swim, cycling or run), distance, duration, the name of the coach, if it is completed by all users and when is last modified. Each session has a number of intervals and each interval has the following information: description, level, duration.
- FR04 - Deleting sessions. An administrator can delete all existing sessions.
- FR05 - Reports. An administrator can produce a report about all sessions in the system.
- FR06 - Browsing. Athletes are able to see their previous and current sessions. But to view only a new session specified by a coach and see all details about it.

They can see information about their all session performance and will be notified about new session information regarding their tasks. Coaches are able to see details about the athletes and sessions as well.

3. Development Process

3.1. Introduction

In the process of choosing the right life cycle model for the project I found out that the waterfall model is not suitable, as it is intended for large projects with very clear, constant and rigid requirements.

I found out that using the prototype process model is attractive in some aspects, as the product is supposed to be an interactive system and it might be quite appropriate to test the prototype somewhere in the middle of the work, thus getting valuable information about possible improvement. Moreover, according to the academic plan, I was supposed to make a demonstration at the middle stage of development, so building a core, but functional product would be required anyway. I decided to use prototype model for developing my application.

I intended to develop a prototype webpage that is completed with main features only needed to get it functional, but which is lacking many minor inessential details. My intention was to perform some testing on the user's side in order to test if the product is fully functional and easy to use from the user's perspective.

3.2. Building a prototype page

The prototype page features the following functionalities:

- User login/logout (coach or athlete or administrator can do it already)
- Post or Get Requests about all user actions and his sessions using JSON (2b) , AJAX (2d) and PHP(18)
- Database tables
- Design of the web page (it will be improved over time)

3.3. Completing the product

Finalization of work - creating a finished product with a full set of functionalities and features needed for trouble-free operation. I have used the following implementation tools in this project:

- Programming language – PHP 5.5.0 (18)
- Server – XAMPP version 1.8.3 (24)
- Web Server – Apache 2.4.4
- Ajax requests - JQuery library 1.9.1
- Database management – phpMyAdmin 4.0.4.1 with database MySQL 5.6.11 (2g)
- Program development environment – Adobe Dreamweaver CS5 version 11 (25)
- Debugging software – Firebug 1.11.4 (26)
- Unit Testing tool – QUnit 1.12.0 (27)

PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. It seems to be the fastest language in comparison to other programming languages. The performance is great, especially when we have large interactions with the database. It is an open source tool, thus the user is free to remodel or recode it. One of the advantages of PHP is that the source code is shipped along with it. Supports various platforms, i.e. can be installed on almost every operating system, like Windows XP, Linux, etc. An additional point is that its syntax is quite easy to code. I have used PHP because I am quite familiar with it and it is easiest language to work with and execute MySQL

database queries. I have worked with the other choice for database sqlite just once and I do not like it at all.

For the server I can say that the latest version of XAMPP is very quick and simple to install and operate. You will need just to download, extract and start. Although initially it has been intended for use as a development tool for testing, later it has become popular as a tool to actually serve web pages. Inside it there are phpMyAdmin control panel for our MySQL databases and Apache Web Server which is already configured to start using it.

I chose to use MySQL, as it is a free open-source relational database management system, which runs as a server and has a multi-platform compatibility. It is good for web applications, integrates nicely with PHP, and its community server is free. It is reliable and easy to use, and especially suitable for small to medium-size organizations that do not plan to generate a lot of traffic. MySQL is updated daily, therefore it is scalable and up to date. It is the right choice when looking for scalability, performance and consistency.

Development part is fun and fast when I use my favourite Adobe Dreamweaver which I am using since it was owned by Macromedia. Excellent feature of this web design and development application is its impressive site management abilities. With the help of this tool it is possible to create entire websites and manage easily links, files and directories via its interface. I could use something like Notepad++ which is good option as well, but I am not used to it.

Program debugging is essential for every web or non web application. One of Firebug's advantages is the feature of organizing and displaying errors as soon as they occur. Another Firebug's strength is the HTML editing and inspecting. The "View Source" window ensures live access to the HTML code of your website. Firebug displays the latest HTML script of the webpage which is inspected.
With Firebug the Web developer can edit and revise the CSS (cascading style sheets). You can preview colors, fonts and images. Any change to a CSS Web element appears straight away and is easily reversible. Firebug is wonderful tool for debugging and fixing errors, instead of using Google Chrome console or Firefox integrated debugger console.

For testing I used a very nice, powerful and easy to use JavaScript unit testing framework called QUnit. It is used by the jQuery, jQuery UI and jQuery Mobile projects and is capable of testing any generic JavaScript code including itself. I have used it to test if a member can be added to database.

4. Design

4.1. Overall Architecture

I have used a 3 tier architecture model that consists of three layers:
Presentation, Application and Persistence.

4.1.1. Presentation layer

It is the way an application is shown on user's browsers. At start this layer makes request to the server which returns the whole HTML page. It is HTML 5 webpage which can be accessed from all known browsers both on desktop or mobile devices. The page is styled using CSS 3 style sheets technology. In addition I have used JQuery for making requests to the application layer and JQuery User Interface libraries for better look and feel tabs capability. At last version I stopped using JQueryUI dialogs and started to use a plugin, which makes logs or alert messages, called Alertify. It is quite good and can show fast or persistent error, log or success messages.

The layer relies on JavaScript and Ajax calls at client-side to server-side (database) via HTTP asynchronous requests and using returned information for making sessions user interface table and other parts of HTML appearance.

Browser compatibility:

- ✓ IE 6.0 +
- ✓ Firefox 3.0+
- ✓ Safari 4.1
- ✓ Opera
- ✓ Google Chrome

4.1.2. Application layer

It is the Web server and the PHP scripts which operate with all requests received from presentational layer (webpage interface). Usually it returns JSON data objects or arrays which then are used in presentational layer via JavaScript. In the file **enter.php** there are many IF statements which are executed if the specified **post** or **get** request with appropriate parameters is received. In these if statements there are queries to the persistence layer (database server) which are executed and data is stored or taken out. When we want to take something with a request the data from persistence layer is converted to JSON object or array or pure HTML 5 code and then send back to presentation layer (browser user interface).

Admin can make a report - FPDF (19) is very useful PHP (18) based class script for free generating PDF files without using other libraries. With a little investigation I found out that some of the scripts in the website can be used to create nice and easy to read reports not only for all sessions but in the future for example just completed ones, added last month or by a specific coach of course with all intervals from the other table. So I saw this WriteHTML script by Clément Lavoillotte and then using it I made really nice PDF output feature. And if administrator goes to make reports tab and click make reports AllInfo.pdf is created with all sessions in the system.

4.1.3. Persistence layer

This layer is the part of the system where we store our information and data i.e. the database in the database server. I am using MySQL database with phpMyAdmin control panel which has nice user interface and is easy to maintain and administrate.

Here is the notation representing the relational schema between all tables:

Users {id, username, password, firstName, lastName, age, coach, trained_by*, email, picture}

Sessions {id, description, type, distance, duration, by*, allcompleted, time_added}

Intervals {session_id, description, level, duration}

TakeSessions {user_id*,session_id*, completed, whenis}

There are UML and Use case diagrams in screenshots folder.

This is data model design:

- Table name: "Users"

Column	*Datatype*
id	Int(3)
Username	Varchar (20)
password	Varchar (30)
firstname	Varchar (20)
lastname	Varchar (30)
age	Int(4)
coach	Tinyint(1)
Trained_by	Varchar (20)
Email	Varchar (25)
picture	Varchar (50)

This table is includes one administrator and all coaches and athletes. If coach field is 1 – then user is coach, otherwise user is athlete and trained_by field is the username of his coach. Email and picture columns were added in the latest version.

- Table name: Sessions

Column	*Datatype*
Id	Int (3)
Description	Varchar (255)
Type	Varchar (20)
distance	double
duration	Varchar (20)
by	Varchar (20)
allcompleted	Tinyint (1)
Time_added	timestamp

All sessions created by the coaches that was different in every version. First I thought a session will consist of all three disciplines and there was distance field for each one of them as well as starting and ending time of the session. In second version I removed all distances and inserted just one distance. In the latest final version I removed starting/ending time fields and added duration and allcompleted field for checking if session is completed (value is 1) by all athletes who had to do it (when a session is done, completed field from takesessions table for the user is set to 1 and then a query is executed to check if all fields with the same session_id have completed set to 1 and finally if this is true allcompleted for the particular session is set to 1). Time_added shows when the session was added by the coach.

- Table name: Intervals

Column	*Datatype*
Session_id	Int(8)
Description	Varchar (255)
Level	Int (3)
Duration	double

- Table name: TakeSessions

Column	*Datatype*
User_id	Int (3)
Sessions_id	Int (8)
completed	Tinyint(1)
whenis	timestamp

Session taken by a user_id and if is completed by him(completed was added later after mid-demonstration because I thought that will be a good way for the coach to know who has completed a session and whenis was added to show when the specified session is completed by an athlete).

4.2. User Interface Design

In the beginning the web site design was not very well structured. There was different header and just tables with basic information about sessions and user details. Some things were removed and new tabs menu was placed for better look and feel for the user. I changed the header I tabs a couple of times, but finally I got it right.

I have added two CSS 3 (2e) style sheets depending on what device you are using to login. First one is for regular desktop computers/laptops and the second one is for mobile devices with max width set to 480 pixels. Most mobile phones have a device-width of 480px or lower, including the popular iPhone 4 (with device-width: 320px), despite it technically having a 640 x 960 resolution. This is due to iPhone 4's retina display, which crams two device pixels into each CSS pixel on the screen. This is true for the IPad 3 as well its reported device-width is 768px just like its predecessors, even though its actual screen resolution is 1536px x 2048px. Mobile view from IPhone (20) website simulator is showing Mozilla Firefox but if you enter the product from real mobile device at the bottom if detected will be displayed your device browser, version and operating system (on my Nokia 500 I see Opera 12 on Symbian). The same is with IPad() simulator webpage which shows how the project will look like on IPad device. See screen shots folder for images of iphone sessions, ipad coach and ipadlook.

When clicked My Sessions tab calls a function which iterate trough JavaScript sessions array and appends table rows for each session to the table. On mobile phone the table is shown like list-item which is vertical and is not going left or right to stretch the screen. On a personal computer or laptop the table is placed nice horizontally in the center of the screen. Any new sessions for an athlete are with red background and an icon new in front.

My Profile tab contains your information and picture. There are buttons for changing your username, password or email after you are registered by the administrator. Also you can browse a picture from your device file system and upload it for your profile picture.

The chart I am using is Google Area Chart and is created when you click to view a session. I have added a chart for each session which contains all intervals from a session and a sound which encourage the athlete to work hard. Something that I could do was to make more sounds depends on interval Level. Also chart update itself at every 30 seconds and the time of most sessions is 20-60 minutes which can be changed. There is sound at the beginning and at 3 minutes from start of a session then every 3 minutes. Also when the focus of the window is lost it stops the chart from changing.

There was a problem with the chart width on mobiles (my Nokia and my landlord's Motorolla: width was set to 700) and I found that the problem was in makeChart() function. There now it takes the width of the device and if is less than 768 width stays the same but if is more than 768 pixels make set width to 700.

5. Implementation

First of all if you do not have JavaScript (2f) enabled you will not be able to use the website because when the page loads it calls functions in jsscripts.js which is included in index.php and admin.js in admin.php file as well. Anyway you will see a message if that happens. When you try to login the form information is send with post request to enter.php (the file that executes all MySQL (2g) queries regarding our database) and if the username and password are correct session variables are created and you are redirected to index.php or if is administrator to admin.php.

If you are logged in member of the club GetRemoteData (your name) is called and AJAX (2d) post request to enter.php is made which returns JSON (2b) text then using JSON.parse (returned text) function, user object is created (then when we need information about user clicking My Profile tab getProfile () is executed and fill up profileinfo table with user.firstName and so on).

After that in the same function is called makeMenu () which makes the menu tabs depending on whether you are coach or athlete using jQuery (5) appentTo (#menu). Then getSessions (type,name) is called and again using post XMLHttpRequest(23) to return JSON (2b) array and sessions array of objects is made by JSON.parse(returned text). Finally if user is coach getAthletes() function sends post request and returns array again then parsing it to athletes array of objects.

So after page is ready we have 3 post requests if coach (user, athletes and sessions), 2 if athlete (user and sessions) and 1 get request if administrator (returns pure HTML (2h) change coach form). In the end of GetRemoteData(name) element with id whois take HTML (2h) value user.firstName and other element with id tabs calls tabs method for making tabs from div elements. After loading to element with id summary is appended browser, version and operating system of the device using BrowserDetect (3) script at the bottom of the jsscripts.js file. All input submit and reset elements are made to buttons with jQueryUI (4) button method and elements eduration1, duration1, distance, edistance are changed to spinners with .spinner method of the same library. The following JavaScript files are imported in index.php head section: jQuery (5), jQueryUI (4), jQuery Titlealert (22), jQuery Countdown (6), jQuery Timeago (7), Google Charts (21) and my jsscripts.js file.

As I mentioned at the start of the previous section I have used AJAX (2d) for all requests to enter.php (making queries to database) and JavaScript (2f) array of objects as data structures. I used JSON (2b) parser to convert responses from data and arrays text to JavaScript (2f) objects.

If Admin is filled all fields and click on Add Member button validateForm() function is called and if all values of form input elements are correct (example is username more than 2 letters and so on) this Alertify (5) confirm dialog appears. If Admin click Add button addMember() is called and post XMLHttpRequest (23) is send to enter.php with all values of elements from that form taken with jQuery (5) val() function. Then MySQL (2g) insert queries are executed and if there is no error "Successfully added coach or athlete" is returned in the element with id feedback.

If Admin intends to add an athlete trained by coach he should not click on CoachMember check box button. This is a snapshot of admin, who enters all parameters. When the username field (element with id user) loses focus, the function check_username() is called. It checks the field, and if it is bigger or equal to 3 characters, the element with id loading is shown, and the value of Username is sent with jQuery (5) post method. After that a MySQL (2g) query is issued to the database in order to check if the username already exists or not. A message appears in return – an element with id error or id success. If an error is shown username is blanked and focused.

If Admin wants to add a coach he needs to click on CoachMember check box button. This actually is a check box with id coach which is transformed to a button with jQueryUI (4) button method. When you click it p element with id trainedp is hidden or shown using jQuery (5) slideToggle() method.

After Admin clicks on Change Coach button, doYouWant() function is called where it checks if you have picked out other than your current coach. Then to change a coach for an athlete you have to click Changeit. This jQueryUI (4) dialog appears and then changeCoach() is called. Inside of it the parameters we are going to send using post XMLHttpRequest (23) are (newcoach="usernameofnewcoach"&radio="idoftheathlete"). And in return we receive in the element with id feedback "Successfully changed coach. Lincoln Burrows 's new coach is Dobromir Kolev". GetRemoteData() is called after 300 milliseconds and makes get XMLHttpRequest (23) which returns new refreshed changeCoach form.
You should change line from 5 to 10 of enter.php .This picture is after you clicked on Make report and View pdf is link to a file that was created just few moments ago.

Admin removes member- this is how looks admin panel with all 5 tabs. When one of the two input buttons is clicked removeMember(0 or 1) function is executed. jQueryUI (4) dialog says "Do you want to remove this member" and then if you press Remove a quick jQuery (5) post request is processed with the id of the user in database. Two "Delete from Users and Takesessions " queries are completed and if no error occurred "Successfully removed member" is returned to element with id feedback and Alertify (28) success log message is displayed on the right bottom part of the screen. Finally an alert appears saying "which id was removed" from database.

Admin removes all sessions if click on Remove All. At the moment it calls removeSessions() function which is just two lines of code using again jQuery(5) post:

```
var params="removesessions=yes";
$.post("enter.php",params, function(response){ alert(response);});
```

Now this is complete wipeout of all the tables using Truncate table sessions, intervals, takesessions. And I put confirm dialog message so right after clicking on it you have to click remove All and then everything inside the database except Users table is deleted. In the future probable can be done with more buttons for removing only completed sessions or last six months sessions.

Admin – when wants to click Add Member validateForm() is called to check if a field is not null, left empty or less than 3 characters and if email field is true valid address using checkEmail(pattern) function. Paragraph with id feedback is showing you a message and the first wrong field is focused. If all are valid Alertify (28) confirm dialog appears saying if you want to add this member and if you click Add it then the member is added.

Member (coach or athlete) after clicking on MySessions tab sees Alertify (28) log message – he does not have sessions yet. (sessions array contains zero objects).

Member of the club can change his **username, email, picture and password** - for username queries have to be executed to change in every coach field in users table. In their profile they see nice picture and all info about them. The age information when added by admin is now year of birth and when shown on profile page as age is calculated in JavaScript (2f) by subtracting year of birth from current year

When coach tries to add a new session validateFields() method is called to check for validation of all fields and message appears if some of the fields is left empty or has wrong value(for example duration or distance must be numbers). When coach pushes Add Session button, the Alertify (28) user interface alert appears. If coach clicks on Add it then we take all the values of all input elements in form with id addsession, all textarea elements and all select elements. We use the jQuery (5) function serialize() - each element's name is equal to its value – this represents a text in a certain order, with sign & between the single elements ("Encode a set of form elements as a string for submission"). An example of the parameters in this exemplary manner is "addsession=user.username&type=type". We serialize the parameters and send them with post XMLHttpRequest (23). And then in enter_php we assign all the values to variables and make MySQL (2g) queries "insert into sessions table", "insert into intervals", "insert into take sessions". In case of error on each of the queries, an explanation of the mysql_error appears back in the element with id success and Alertify (28) success as well. If there is no error, it appears back "successfully added session".

Coach has clicked on Edit Session input button editSession() is called. It does almost the same as when you add new session function. First check function validateFields() if any of the form fields are not valid. If then press Change it we serialize all form elements and send all parameters with jQuery (5) post method to enter.php. In return the text responded is inserted into paragraph with id edittest. Finally call getSessions(type,user) to update sessions array of objects.

Coach clicked on second tab sees Alertify (28) log message if has no athletes he is training yet.

When Coach wants to edit a session he clicks on Edit button which calls addTab (array position in sessions). There it sets all values of elements with information (for example sessions[i].description) then calls addEditIntervals () which makes intervals like the session we want and put values inside new created elements. Also takenBy (sessions[i].id) is called to make check box buttons of all athletes that take this session. I have not put checking on add new session and edit session but in the future it must be done.

If a coach wants to view a session he clicks View button which calls seeSession(i). Then he sees the chart and all other information about the particular session. In addition there are three buttons on top which are Next , Previous and Edit. Next or previous draw the chart and all text for next or previous chart with makeChart(x) and x is sessions array of objects number.

Coach's sessions where if you mouse over to a row Description you will see it in full text and Duration is shown in a pretty look with getNiceDuration(sessions[i].duration). Column Added is represented with abbr element with class attribute timeago inside each row and body of the element getNiceDate (newDate(sessions[i].time_added)). In the end of coachSessions() or athleteSessions() is called jQuery Timeago (7) method timeago() for all elements with class timeago (all abbrs) and that is how the date is converted to timestamp(like 23 minutes ago).

Every coach can see all athletes trained by him- when clicked on the second tab showAthletes () is called and populates a list with names, age and pictures in id myathletes. The data for each athlete is taken from athletes array of objects (example is athletes[0].firstName).

Coach's sessions when all users have completed a session has green background – coachSessions() function is called when you click on first tab and there adds to the table with id mySessions all information from sessions array of objects (for example sessions[0].distance) to it as rows using jQuery (5) methods empty() and appendTo(). When a session is not completed by all users who have taken it, an attribute class complete is added to the row (red background).

Coach John adds a new session. This is New Session tab. There are placeholders attributes on the textarea and select elements. You can add more intervals and remove existing ones but you cannot delete the first one. And you can choose from the check boxes which are changed to buttons using jQueryUI (4) library and clicking on one of them will choose an athlete. Clicking on Add Session will show the dialog which is explained at the start of this section.

Athlete has completed session and that's why there is no View button.

When an athlete clicks start button to start a session he sees this Alertify (28) confirm dialog message. If Start is clicked an anonymous function is executed where start button is hidden, tabs are disabled, seconds of the whole session are calculated (variable per) and audio tag with id sound plays (.play() method) Come on file either in ogg or mp3 format (there are three ogg and three mp3 sound files in sounds folder). The reason that I have two files for each sound is that each browser supports some type of codec for the audio tag. I have just 3 sounds because that is what I came with but there can be for example 10 sounds for each individual level so when is played to be the same as current level of exercise. Here are currently supported codecs:

Browser	Ogg Vorbis	MP3	WAV
Firefox 3.6+	✓		✓
Safari 5+		✓	✓
Chrome 6+	✓	✓	
Opera 10.5+	✓		✓
IE 9 (beta)		✓	✓

The src attribute is set using getAudioType() function which depending on the browser checks which type of audio file to choose. Then the chart with id good is shown as well as countup and countdown elements and the other tags for intervals and description. Elements with ids short and shortly call jQuery Countdown (6) plugin methods to set since and until attributes of countdowns assigned to them. On expiry of the session function liftOff () is called. If Cancel is clicked nothing happens.

When athlete clicks on MySessions tab see his sessions. In the top of the browser there are jQuery TitleAlert (22) which is started with interval of second and a half (1.5) (like messages on Facebook when someone sends something to you) and he sees (1) New Sessions alert every 1.5 seconds.

Athlete before starting a session he has clicked View button and startSession(x) is called. Title Alert (22) is stopped and Google Chart() is made with makeChart(x) function. There is a progress bar as well which is updated constantly. Element with

id momentdesc changes it's body to sessions[x].description, sessiondist changes to sessions[x].distance and on table with id mySessions is called empty().

Athlete is in the middle of a session. There is a Pause button which pauses the countdown and countup elements and with this changing the chart and everything. Then of course clicked again it resumes the session rithm. After Start button is clicked and then Start as well countup and countdown elements are updated every second using watchCountup () and watchCountdown () functions. In the first one variable soundplayed=180 which is when the sound come on will be played and then every 30 seconds is played again. When the time is exactly end of an interval it changes id intervaldesc to sessions.[x].intervals[x].description and update intervaltitle with the next interval information. WatchCountdown updates every 30 seconds the map calling nextChart() function which updates chart axes with constant speed of half minute and draw it again:

```
options.hAxis.viewWindow.min += 0.5;
options.hAxis.viewWindow.max += 0.5;
```

On the end of session liftOff () is called. Inside of it components with ids short and shortly are destroyed using jQuery Countdown method "destroy" and sendFeedback () is executed. Sound time_up is played, tabs are enabled, chart is shown in first state and a jQuery (5) post request is send to database changing field completed to 1 for specific username and session and then checks if all users have completed set to 1 changes allcompleted to 1. It also changes field whenis to timestamp of the current time so then when coach wants to view the session he sees that specific user has completed the session 20 minutes ago. Here are sent parameters:
"completedby="+user.id+"&sid="+sessions[whichchart].id

6. Testing

6.1. Overall Approach to Testing

I decided to use the dynamic testing strategy. I suppose it would be the best way of testing the system, as I used partially the prototype process mode in the development of this piece of software. My intent was to perform some testing on real users somewhere in the middle of the development process in order to record the results obtained. This could help in analyzing the faults discovered and assessing if the initial requirements are met adequately. Thus I would be able to decide if the product is suitable and easy to use for the target audience and to outline possible ways of troubleshooting and improvement.

6.2. Automated Testing

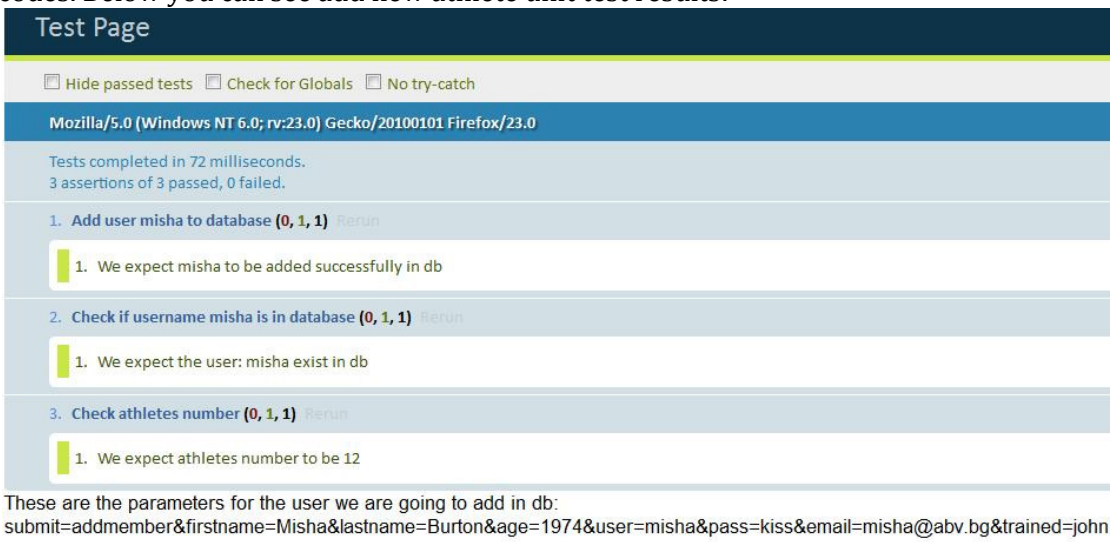
Automated testing is proven to be quite effective. It involves building automated tests needed to carry out some repetitive tasks which could be quite laborous to perform manually, or their manual performance would lack efficiency.

6.2.1. Unit Tests

Unit tests usually involve testing of every part of a given program to verify if it is working properly against a strict set of requirements. In order to test a component of my application, I needed to check the functionality of some portions of the code against a certain function to verify if they are acting as expected in various conditions. I decided to perform three unit tests in page **testing.php**. which adds a new member to the club

database using AJAX post request.

I performed JavaScript (2f) test using QUnit framework for testing generic JavaScript codes. Below you can see add new athlete unit test results:



Next image is when I tried to run it again with the same parameters and it fails the first assertion of course because a user with same username is already added:

The unit testing I carried out is supposed to facilitate the integration testing and integration itself, as I used the bottom-up testing approach, i.e. first I checked the parts of the system and then the whole system.

6.2.2. User Interface Testing

This is the process of testing the user interface to check its conformity to the specified requirements. For the purpose test cases must be created. In the simplest case a test case is the input and the expected result. I used a software allowing me to record all user actions in video files in order to show you the user interface point of the results obtained. I filmed video films for Add New Session, Edit Session and Add member functions of the system. All three are in directory called **testing**.

6.2.3. Test specification and results

ID	Requirement	Description	Inputs	Expected outputs	Pass/Fail	Comments
1	FR1	Test login	Credentials Girl/good	Screenshot *loggedingirl* is displayed	P	
			name/apass	Screenshot *wronglogin* is displayed	F	
2	FR1	Add member	Enter email **some@abv.bg**	Valid	P	
			Enter **sometext**	Screenshot *not valid email*	F	
		Change Coach	Select new coach and click change	Screenshots *Chcoach* and *schanged*	P	
		Change info	Change details clicked	Screenshot *updated*	P	
		Remove Member	Select member and click Remove	Screenshot *removeuser*	P	
3	FR2	View Session	Coach click view	Screenshot *coachview*	P	
			Athlete click view	Screenshot *athleteview*	P	
4	FR3	Add Session	Entered distance is **5**	Valid distance	P	
			Entered distance **minus 5**	Screenshot wrongvalue	F	
		Edit Session	Entered duration is **10**	Valid duration	P	
			Entered duration is **sometext**	Screenshot *wrong2*	F	
5	FR4	Remove All Sessions	Remove All clicked	Screenshot *removeall*	P	
6	FR5	Make Report in PDF format	Make report for all sessions clicked and then viewPDF	Screenshot *pdfreport*	P	
7	FR6	Athlete View Only New Sessions	Log in as user **ssa** with pass **pound**	Screenshot justnew	P	

7. Evaluation

In my opinion I managed to make a good looking, easy to use web application.
At the beginning of my development work I stated very clearly all the requirements to my product, namely: I stated very definite objectives: to make a website for triathlon athletes who are away from the club. They should be able to check their sessions in real time and get full information about them. The purpose of creating such a website is to facilitate the interaction between coaches and athletes, so that both groups could be able to keep track of the training schedule. All the functional have been also defined at this stage, namely: three types of users – athletes, coaches and administrators; set of available sessions and functions with the sessions – deleting a session, adding a session; producing reports about all sessions or sessions for a certain period; browsing – opportunity for the athletes to see previous and current sessions, all session performance and new session information regarding their tasks, for coaches – to see details about their athletes and sessions as well.

As to the next step – finding a design solution, I think that I made the right decision for this type of application choosing the prototype process model. I built a simplified product in the middle of the development process that lacked some features and functionalities, but was functional enough to pass testing.
As to my choice of technologies and tools, I tried to select the most appropriate of them. Of course, my choice depended also on my knowledge and familiarity with technologies, tools and programming languages.

I decided to use the PHP as it is a general-purpose scripting language that is especially suited to server-side web development where PHP runs on a web server. The use of the CSS provides more precise formatting, separation of HTML content from appearance, it saves time, ensures easier site maintenance and also web accessibility.
For testing I used a testing framework called QUnit. With QUnit, you can write tests for each functionality of your code, and if all of these tests are passed, you can be sure that the code will be bug-free. I used MySQL as database, as it is a free open-source relational database management system which is suitable for web applications, integrates nicely with PHP; it is reliable and easy to use as well.

For the XAMPP server I can say that it was the right choice. It is very quick and simple to install and operate. With Ajax requests, you don't have to reload an entire page just to handle username/password verification: it makes your web application more responsive.

If I can start again maybe would use PHP object oriented classes structure or JavaScript one. Also the three types of users implementation should be in different files perhaps. I think the product purpose which is just for indoor training is a drawback, but in the future the website can improved by adding more sounds and things like that to be usable outdoors as well.

I suppose this web application would serve the purposes and would satisfy the needs of the targeted audience – triathlon athletes and coaches – in their everyday activities. It would be an excellent tool serving to facilitate their communication, interaction and keeping track of the latest events. Such type of a product is very useful for the modern life that became very dynamic, busy and pressured. I improved substantially the latest version of this software product in order to better suit it for use on iPods, iPads, mobile phones and other similar devices on Android and iOS platforms.

Appendices

A. Third-Party Code and Libraries

Using JQuery UI and libraries, JQuery Countdown plugin, JQuery Timeago plugin, JQuery Titlealert plugin and FPDF libraries for making PDF files and code by Clément Lavoillotte for MYSQL report in my enter.php file. Using Firebug for debugging and testing the page.

B. Code Samples

The code HTML2PDF by Clément Lavoillotte
http://fpdf.org/en/script/script42.php.

Annotated Bibliography

1. A Wikipedia article about a type of three-sport athletic competition http://en.wikipedia.org/wiki/Triathlon
   It gives really nice information about this sport.
2. The world's largest web development site
   a. http://www.w3schools.com
   A lot of examples about timeouts and interaction.
   b. http://www.w3schools.com/json/default.asp
   JSON tutorial – What is JSON and what it can do.
   c. http://www.w3schools.com/xml/default.asp
   XML tutorial – How to use the XML.
   d. http://www.w3schools.com/ajax/default.asp
   Ajax tutorial - AJAX is the art of exchanging data with a server and updating parts of a web page.
   e. http://www.w3schools.com/css/default.asp
   All about CSS styling
   f. http://www.w3schools.com/js/default.asp
   JavaScript scripting language of the web
   g. http://www.w3schools.com/php/php_mysql_intro.asp
   MySQL is the most popular database system used with PHP
   h. http://www.w3schools.com/html/html5_intro.asp
   HTML 5 – New Standard
3. Browser detect script
   http://www.quirksmode.org/js/detect.html
   A useful but often overrated JavaScript function - the browser detect.
4. jQuery user interface interaction http://jqueryui.com/
   jQuery UI - a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.
5. jQuery documentation
   http://docs.jquery.com/
   A useful resource giving information about jQuery documentation, tutorials, bug tracker, plugins, UI, libraries
6. jQuery countdown plugin http://keith-wood.name/countdown.html
7. jQuery timeago plugin
   http://timeago.yarp.com/
   Timeago is a jQuery plugin that makes it easy to support automatically updating fuzzy timestamps.
8. https://latitude.google.com/latitude/b/0 google geolocation
9. http://www.mozilla.org/en-US/firefox/geolocation/ info about geolocation
10. http://dev.w3.org/geo/api/spec-source.html W3C geolocation api
11. http://www.cutepdf.com/products/cutepdf/writer.asp How I did this pdf file
12. https://maps.google.com/ Google maps
13. http://docs.phonegap.com/en/2.2.0/cordova_geolocation_geolocation.md.htm Apache Cordova Api Geolocation
14. TrainingPeaks http://home.trainingpeaks.com/software-for-athletes.aspx
15. Triathlete http://triathlete-europe.competitor.com/2010/03/10/starting-

triathlon-vary-intensity-levels
16. http://training.220triathlon.com/fun-and-friendly-races
17. England athletics http://www.englandathletics.org/homepage.asp
18. A Wikipedia article about the PHP scripting language https://en. wikipedia.org/wiki/PHP
19. FPDF 1.7– making a free PDF by Olivier PLATHEY http://fpdf.org/en/script/script42.php HTML2PDF by Clément Lavoillotte
20. Mobile IPhone simulator http://iphone4simulator.com/
21. Google Chart https://www.google.com/jsapi
22. jQuery TitleAlert http://heyman.info/2010/sep/30/jquery-title-alert/
23. XMLHttpRequest http://www.w3.org/TR/2012/WD-XMLHttpRequest-20120117/
24. XAMPP http://www.apachefriends.org/en/xampp.html
25. Adobe Dreamweaver http://www.adobe.com/uk/products/dreamweaver.html
26. Firebug v.1.12.0 https://getfirebug.com/
27. QUnit - Unit testing v.1.12.0 http://qunitjs.com/
28. Alertify v.0.3.10 http://fabien-d.github.io/alertify.js/
29. IPad Peek http://ipadpeek.com/
30. CSS Queries http://www.javascriptkit.com/dhtmltutors/cssmediaqueries2.shtml