



MAJOR PROJECT FINAL REPORT

Interactive Multimedia Design (2016)

Student: Ariel Piecha – B00619406

Mentor: Richard Davies

PSG: 7

Sporty: <http://46.101.94.209/>

ACKNOWLEDGEMENTS

I would like to thank Richard Davies for his mentor assistance on this Project. Richard has provided useful help by giving some great ideas for different features through the year, additionally he provided a great feedback that kept me on track and gave great advice throughout the project.

Also I would like to thank all the teaching staff on Interactive Multimedia Design course as throughout the years they have provided great material. Without it I would not be able to create Sporty. I am extremely grateful for Course Director Dr. Peter Nicholl who has always been there and works very hard in order to keep the course relevant with new web technologies.

Special thanks goes to my fiancée Jagoda, for her constant support, advice and believing in me and my parents who provided great support over the years.

Contents

ACKNOWLEDGEMENTS	2
1. INTRODUCTION	5
1.1. Project Overview	5
1.2. Aims & Objectives	5
1.2.1. Scope	6
1.3. Work Undertaken Overview	7
1.4. Report Overview	7
2. CONCEPT DEFINITION	8
2.1. Idea Generation	8
2.2. Contextual Research	9
2.3. Requirements Specification	11
2.3.1. Functional Requirements	12
2.3.2. System Requirement	13
2.3.3. Non-Functional Requirements	13
2.4. Paper Prototyping	15
2.4.1. Initial Sketches	15
2.4.2. Refined Sketches	16
2.4.3. Wireframes	16
2.4.4. Style Tiles	17
2.5. Feasibility Testing	17
2.6. Methodology Selection	19
3. DESIGN	20
3.1. UX Design	20
3.1.1. Branding	20
3.1.2. User Journey	22
3.1.3. Responsive Design	24
3.2. System Design	24
3.2.1. Client-Server Model	24
3.2.2. Technology selection	25
3.2.3. Model View Controller	28

	4
3.3. Data Design	28
3.4. Risks	29
4. IMPLEMENTATION	32
4.1. Technology Review	32
4.2. Technology Use	34
4.3. Challenges	35
4.4. Achievements	44
5. TESTING	48
5.1. Test Approach Selection	48
5.2. Test Process	48
5.3. Test Results	49
5.4. Beta	51
6. EVALUATION	52
6.1. Beta Evaluation	52
6.2. Project outcomes	52
6.3. Methodology Evaluation	53
6.4. Plan Evaluation	54
7. CONCLUSION	55
7.1. Report Summary	55
7.2. Project Reflection	55
7.3. Role Reflection	55
7.4. Future	56
9. REFERENCES	57
10. APPENDICES	61

1. INTRODUCTION

This section will cover and explain project overview, the aims and objectives for the project, work that was undertaken and the report overview. Name of the project is Sporty; it is rich internet application for creating sport events.

1.1. Project Overview

Sport is a great way to exercise and meet new people. In England alone there are 15.5 Million people aged 16 years and over that participate regularly in sport activities (Who plays sport?, 2015). Although from statistics we know that the number of people who join sport activates is decreasing with age (Participation Report, 2015) and there are couple of reasons that this might be caused by. Research shows that some people simply loose contact with sport over the years and they just don't know how to get back into it. Some people on the other hand maybe don't have any friends or don't know any people who participate in regular sport activities with whom they could play with. Sporty is going to solve this problem by connecting people together, this web application is going to enable people to create/search sport activates in users area, where they will be able to sign up to event, search for people to play a sport of their choice with and simply connect people who don't know anyone who plays sport but they would like to start participating. Sporty is also going to help regular sport participants to find missing players quickly for team activities.

1.2. Aims & Objectives

The aim of Sporty is to enable people who might struggle to find someone to participate in a sport activity with and to be able to create events in their area, as well as connect people who regularly take part in sport and enable them to search and find missing players and connect with other people.

Project objectives are as follows:

- Create simple but high standard responsive Design and User Interface that will enable young & older users navigate the site easily.

- Enable users to create account where they will be able to create their profile, see events they participated in and make connections.
- Enable users to easily and quickly create new events, where they will provide information regarding the event and its location.
- Enable users to quickly search for events in their area and sign up for them.
- Implement additional features based on the scope and time frames.

Technical Objectives:

- Learn more about the technologies and languages that are going to be used (Frameworks, APIs, jQuery, PHP etc)
- Choose appropriate methodology for the project
- Create a usability survey and test application with users
- Use version control system
- Create strong database relations to suit the project needs.
- Use wireframes to create design of the website.

1.2.1. Scope

Project is scheduled with limited time; deadline is set for end of April 2016, which gives limited time to learn, develop, test and push the project to production stage. Sporty is a university project and the budget is set to £0, it is going to be free to create. Additional costs to purchase domain, host or some API might arise during project. Supply money for additional costs will be provided if necessary. Following resources are required: laptop, desktop and phone for developing and testing, text editor, virtual machine environment for developing the website, Laravel 5 framework, MySQL workbench, adobe suit and different browsers for testing purposes. Main plan is to deliver the project in high quality by using frameworks and solutions that are well recognized in web industry, also there is a plan to organize the work load to be aware of the progress and keep track of it, as well as reflect on it when needed.

1.3. Work Undertaken Overview

Overview of the work undertaken for this project is split into few different phases; planning, design, development, implementation, testing and deployment. Over the period of nearly 6 months a lot of this time was spent learning on how to develop and implement all the features for this project using the Laravel framework. Schedule of the work can be seen in [Appendix 10].

During planning a lot of focus went into research and thinking about how users are going to use the site, things like listing all the necessary functional requirements and non-functional requirements. It was also very important to think about all the features and how they are going to be implemented using different programming languages, libraries or APIs.

Design phase consisted of how to create a great user experience for the user. During design phase things like colour scheme, fonts, images, branding, user journey were looked at. Also system design and how different programming languages are going to be used was looked in a bit more detail.

Development and Implementation phases took the most time. During this time main features were built. Database, back-end and front-end were developed and implemented with hard work development phase was completed, a lot of learning had to be done during this phase.

Testing was split into two smaller phases, a closed beta testing involved only certain people with technical background to have access to a deployed test server and open beta where everyone was welcome to come to the site and give relevant feedback. After that final amendments were made and site was deployed and went live.

1.4. Report Overview

This report goes on and covers different sections for creating this project, you will be able to read about the concept definition, design and implementation. After that report follows into the testing and the evaluation. The final section of the report covers conclusion, the resources and appendices.

2. CONCEPT DEFINITION

This section will dive into the contextual research, idea generation as well as the requirements that are needed in order to complete the project, paper prototyping, feasibility testing and methodology selection for the project.

2.1. Idea Generation

During the idea generation phase, three different ideas were listed and taken into consideration for major project. First idea proposal was a CSS challenge interactive website where web designers would be able to take on weekly challenge in designing something with HTML and CSS only. Every week challenge would change and it would be based on a random generated word. Designers would submit an entry for everyone to see, but no code preview would be shown at that time. People would then vote for the best one. By the end of the week five highest voted entries would have their code available for others to view. People would then be able to compare different styles and techniques that web designers have used and learn from each other. Users would submit the entry by supplying files or an online text editor could be implemented to check for specific language, then website would render the files and display the submission on the challenge page for that week.

Second proposed idea was an interactive card game for children. This project would allow children to learn basic to intermediate mathematic by playing an interactive game. This card game would aim at children aged 3-10, the game itself would have different levels where children would progress and gain experience by solving mathematical problems. The objective would be based on children selecting correct answer, at the start of each level a particular number of cards would show on the board, these cards would act like enemies. Child would have to add up a numbers that show on the cards and select the correct value in order to progress to the next stage, by selecting a correct answer – animation would be played to show victory. Depending on the level, different numbers or even equations would show on the cards. Children would have five lives to complete whole set of stages, after selecting wrong answer live would be deducted. If all lives are lost it would be a game over. By completing the course, children would be rewarded with certificate, which they could print out. They would also unlock another harder level to progress to.

Last proposed idea was a sports website – Sporty - where users would be able to search for people to play a sport with. Idea for a website like Sporty came to mind when looking to play football but none of friends wanted to join in. Struggling to find people and in the end resulting in staying at home. The idea was born to create a website where people will be able to find strangers in their area which want to play particular sport and basically connect with each other and play it. Website would provide search and find feature where user will be able to sing up to an event that somebody else created. Meeting point and where they want to play would be provided on the event page. People would exchange phone numbers or some sort of information so that they can contact each other. Users would be able to create their own profile and provide some useful information about their sport history and the events a user has participated in.

The chosen idea for major project was the sports website “Sporty”. This decision was based on few factors. Firstly, looking at the three ideas list of all the features and the challenges with the implementation was created. First idea turned to be much more complicated than it looked. Allowing people to publish and render HMTL and CSS without the ability for other users to inspect or do a source code seems like a very hard job to do. Also the implementation to check that the user has not submitted a file with some parts written in other programming languages looked very complicated. By reviewing second idea it was realized that it seems to be very design based, creating a website for children would require a great and friendly design but also a lot of detail would have to be spend in creating fancy animations, creating characters and cards in order to keep children engaged. Design is something that is my weak point and I didn't feel comfortable with this idea. Last idea listed a good list of features and a technical challenge that felt double and feasible, also it offered good potential for expanding and adding addition features such as creating a Team or creating a league with friends.

2.2. Contextual Research

This section is going to cover existing solutions and the gap that Sporty is willing to solve by developing this web application.

Currently there is no web application such as Sporty where people can advertise their sports events and enable people to sign up for them. There are however similar solutions like Teamer (The free sports team management App, 2008) or TeamSnap (TeamSnap, 2005). These two websites are the most popular when it comes to sport and people connecting with each other, they mostly concentrate on the team aspects where a Team Coach or a Leader has the ability to contact players about the upcoming games or events, they are able to choose the team and players are able to communicate with each other, team members are able to add pictures and documents to team's profile pages where everyone within a team can look at them.

Teamer is a free sports team management website application that is designed to take out any problems with managing any sports team. It uses web to significantly decrease the amount of time it takes to get the team on the field. Teamer offers few different features, members where you are able to see all the players, supporters, coaches or anyone interested in the team. It offers event notification system that uses email, text messages and mobile app notifications. Users are able to respond to notification to mark their availability for the event, this can also be done in advance. Ability to store all team's documents and upload pictures to the team page. Teamer payments feature offers the ability to collect money of members so that it is easy to track and can be done online.

TeamSnap mission is to build a great product that any parent, coach, team manager, group organizer or sentient robot can pick up and use immediately. They offer similar features to Teamer like home team page, payments, schedules, members availability, photo sharing and messages. Some other features that they include is refreshments notifications, users are able to check who is responsible for bringing refreshments or just give them opportunity to list what they are bringing. Sponsorship feature allows to list all the important sponsors on team's home page. TeamSnap also offers other tracking feature that enables to check who has brought any extra socks, gloves, shoes or any other items.

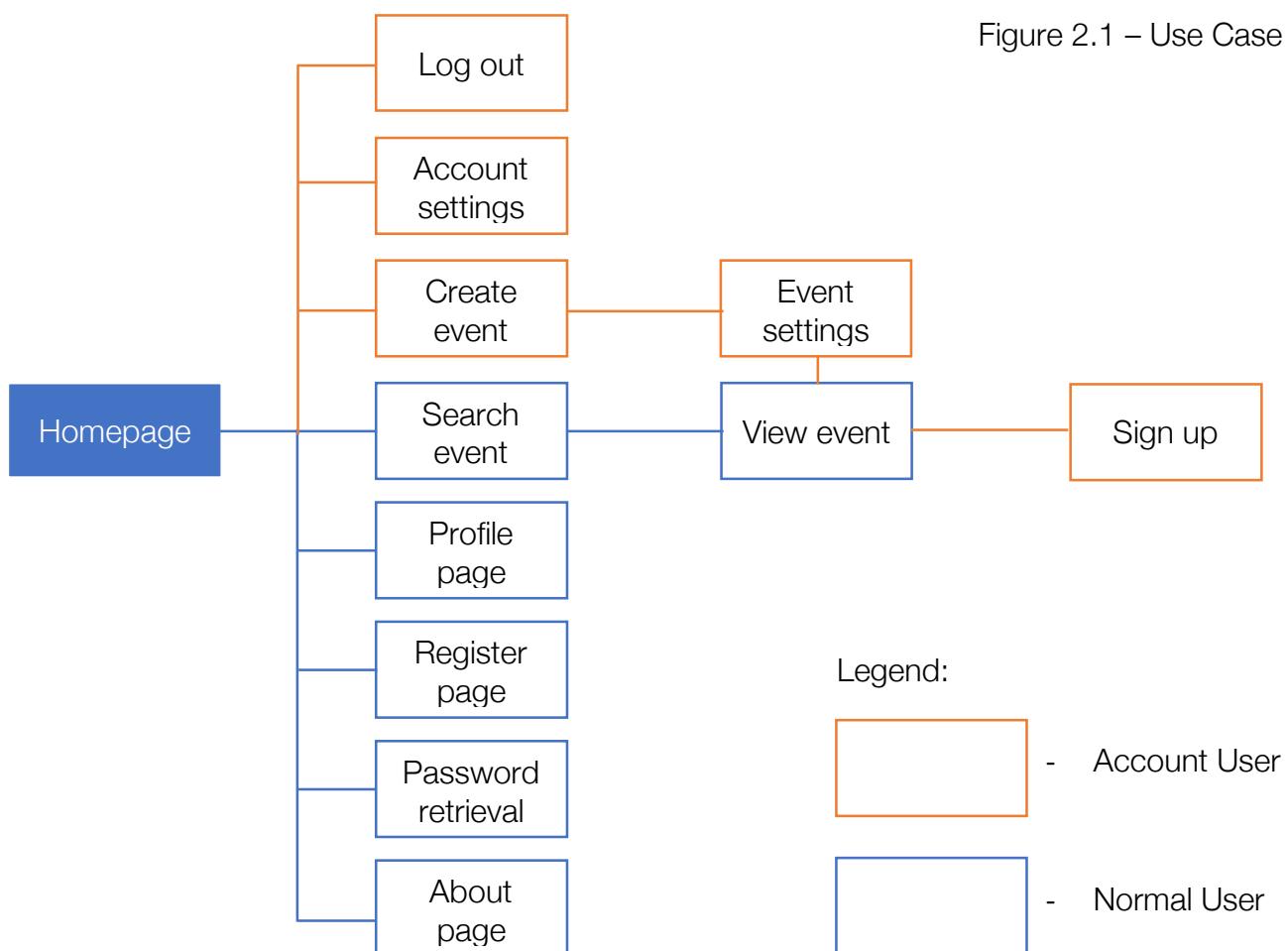
The problem and the gap with these two websites is that a person who is not within a team can't participate in any sport, they are not able to search or create sports events for anyone.

By developing Sporty it will enable people to find events that they always wanted to participate in but perhaps they just didn't feel that they are good enough or they didn't know anyone who plays sport that they are interested in to play with them or a team needs to find a missing player for event. Sporty will enable people of the same skill level to play with each other, it will enable people to advertise sport events in their local area and it will connect random people with each other so that they are able to meet new people and make new friends.

2.3. Requirements Specification

In order to fully know the functionality and the requirements of the project decision was made to create example of use case for a user and account holder. This will help to distinguish functional requirements, system requirements and other non-functional requirements.

Example of use case:



As you can see from figure 2.1 the user is going to have few options from the homepage, they will be able to register and create an account with the site which will enable additional options like creating events and signing up for them. Search feature will be enabled for all the users so that the user can browse through and look for events they are interested in. Some features like editing the event will only be available for the user that created the event. Also user will be able to view other user's profile pages but in order to create one, user will need to sign up and create an account.

2.3.1. Functional Requirements

Functionality requirements are laid out using Volere Snow Cards. Choice to use Volere Template was made in order to create them, this is a well known method for creating requirements and it felt that it will work with project well, because this template method has been used around the world with many different organisations. Here are 3 different examples of functional requirements, rest of them can be seen in [Appendix 1].

Requirement #:	
Priority	1
Description	Allow users to register and create account in web application
Rationale	Enable users to use all features within system when logged in
Dependencies	User needs to provide information in order to create their account
Fit Criterion	Enable users to login to the system, display confirmation message on successful account creation.

Requirement #:	
Priority	1
Description	Create a search bar and search options
Rationale	To allow users to search for events
Dependencies	No dependencies

Fit Criterion	Return a list of events based on search selection, suggest events if no events found
---------------	--

Requirement #:	9
Priority	1
Description	Allow users to create a profile page
Rationale	Other people can view your page and contact you
Dependencies	User needs to be logged in to the system to access account settings
Fit Criterion	User provides details about themselves like: skill level, age, location, sports they are interested in and nickname, email etc

2.3.2. System Requirement

Within Sporty system there are also couple of specific requirements that the system will need to handle, for example the creator of the events need to have different options then a user who is viewing or decides to sign up for the event, here is a list of them:

- Allow event creators to pre-pick teams
- Allow event creator to contact all signed up users for event
- Allow event creator to cancel the event
- Allow event creator to change event location
- Allow event creator to confirm signed users
- Allow event creator to confirm the score / teams for the event.
- Allow signed user to drop/resign from the event
- Allow signed user to rank the event after it is finished.

2.3.3. Non-Functional Requirements

Non-functional requirements are split into three different tables, Design, Performance and Usability. Full list for all the tables and the requirements can be seen in [Appendix 9].

When looking into design it was noted that the website needs to have clean and simple design so that the users can navigate easily on it. Website needs to have a good colour scheme so that it is attractive and looks professional. Sporty needs to be responsive so that is it accessible on different devices. Application needs to work on different major browsers. Application should use design icons to minimise text on screen.

Design		
Requirement	Priority	Description
1	1	Create a clean a simple design to navigate the website easily
2	3	Website needs to have a good colour scheme to make it look professional and attractive
View rest is [Appendix 9]		

With the performance it was noted that the website needs to run smoothly with no errors or crashes. Sporty needs to be available for use online at all times. Website should be fast, reliable and handle lots of users at the same time. Application needs to be easily installed, maintained and be transferable as well as extendable.

Performance		
Requirement	Priority	Description
1	1	Website needs to run smoothly with no errors or crashes
2	1	Website needs to be accessible online at all times
View rest is [Appendix 9]		

When it comes to usability, few requirements were noted. Website needs to protect all user's data. Website needs to take regular backups of data and be secured. Sporty needs to have

good accessibility to remove the barriers of interaction or access to the website by people with disabilities.

Usability		
Requirement	Priority	Description
1	1	Website needs to protect user's data
View rest is [Appendix 9]		

2.4. Paper Prototyping

During the early stages of the project it is good practice to draw sketches and different ideas into the paper especially when designing a website. Initial first step was to think about how the website will work and how the user will navigate through the website. Simple site map was created, which can be seen in [Appendix 2].

2.4.1. Initial Sketches

First sketches for the application were drawn using 4-ups method. This is a well known process in design industry, the ideas get drawn on paper and designer can see which design would suit the project best. Sporty has couple of main pages like homepage, list of events, event page, and profile page, for all of these pages' sketches were drawn using 4-ups technique, you can view all of them in [Appendix 3].

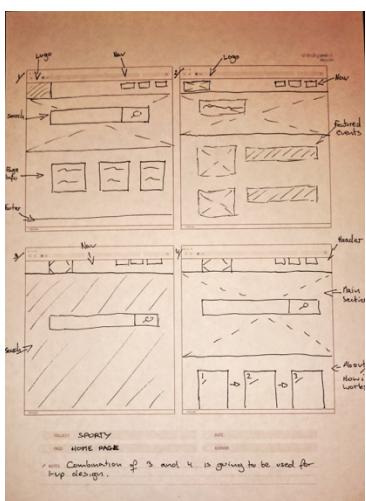


Figure 2.2 – 4-ups for homepage

2.4.2. Refined Sketches

After initial sketches were drawn, next process was to pick one of the design from 4-ups and create a 1-up design with more detail which would give a closer view to the final design of the page. This has helped to see where different elements on the page would go and it helped to imagine the position of main functionality like search bar, profile information, details for the list of events. Furthermore, start to think about the colour scheme and the fonts that would suit the project was made. All of the 1-ups can be seen in [Appendix 4].

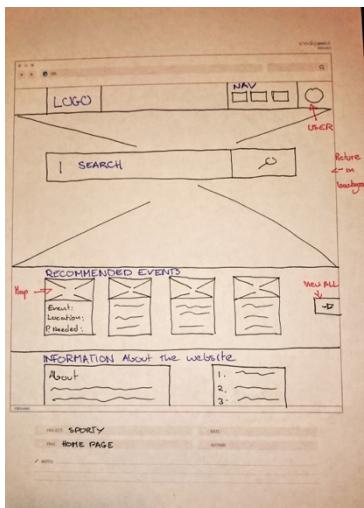


Figure 2.3 – 1-up for homepage

2.4.3. Wireframes

Last process of paper prototyping was to do even more detailed mock-up of the designs; decided to use wireframe technique to draw final sketches. This method allowed to tidy up the design and think about creating a simple user experience. Users can get confused very easily so having a logical and simple design helps to achieve good user experience. While creating wireframes thoughts about the size and width of the elements was made so that the information that will go inside would have enough space to fit into it. Wireframes should reflect the final layout of the page so it was crucial to think about small details and all of the information that the pages want to display. Wireframes for every page can be seen in [Appendix 5].

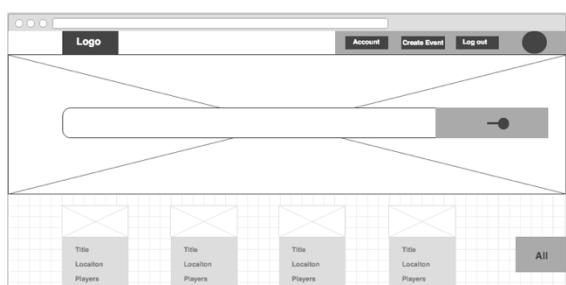


Figure 2.4 – Wireframe for homepage

2.4.4. Style Tiles

Style tiles allow to create a mock-up of the fonts and colour scheme for the website, and the idea behind it is that it is good to see the design style early on in the project so that reflection can be made on it later on if it doesn't suit well. This allowed to visually image the pages by combining style tiles with wireframes, two different version of style tiles were created and decision was made to chose one, which can be seen in the Figure 2.5. Rest can be seen in [Appendix 6]. Decision was based on how different colour elements will stand out from the main colour on the page, plan is to have one colour for one sport. Feeling that the chosen style tile is also more pleasant for the eye and it looks more professional.

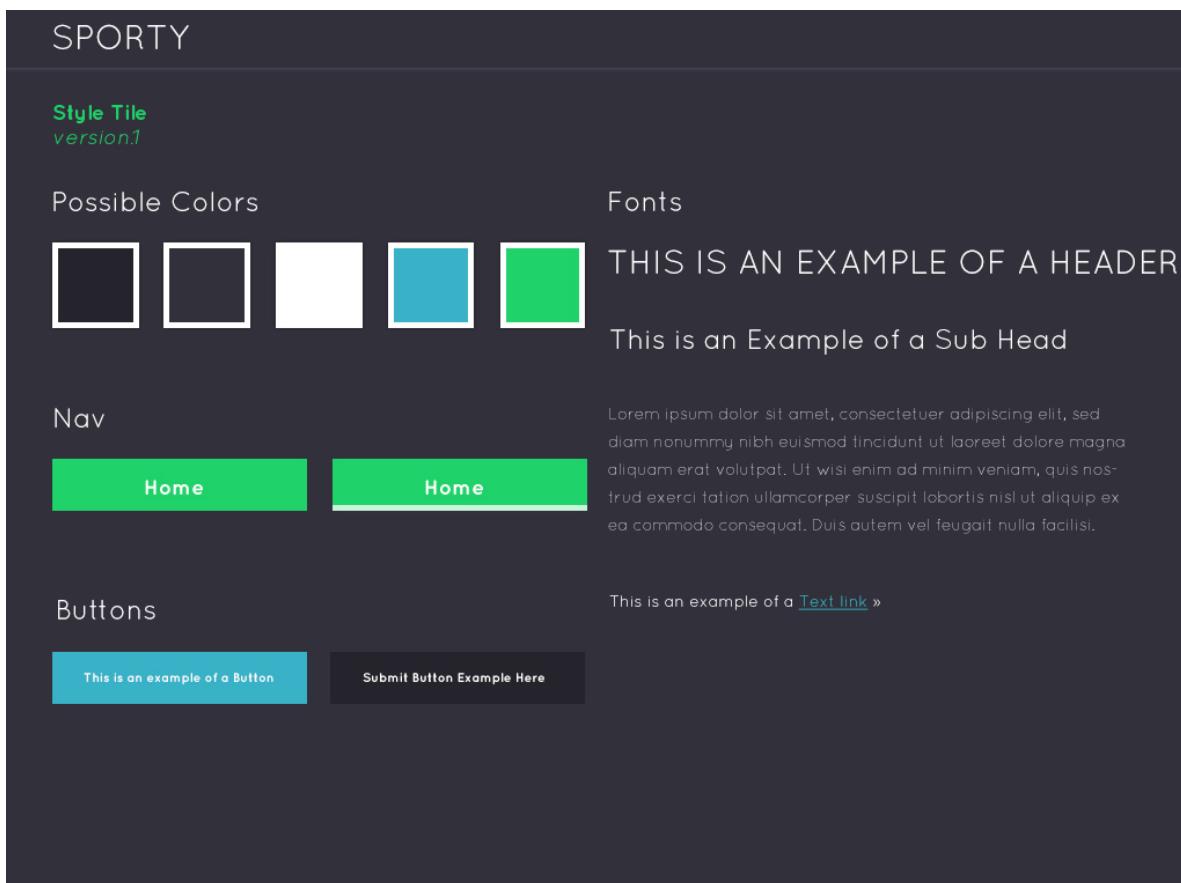


Figure 2.5 – Style tile

2.5. Feasibility Testing

Feasibility testing is an approach to test if the initial product is viable or not. It gives a reasonable check on the time frame, determines if the technology is feasible and if the project can be finished with the resources available.

In order to test some of the technical requirements a prototype was created. Having no technical experience in using Google Maps API it was a good idea to check if it is suitable and well documented as the project relies on it heavily. The main purpose of the prototype was to get the basic logic of the event creation form and how it is going to save location of the clicked marker on the map.

Feasibility Testing Result:

Very basic website was created using HTML and CSS which integrated Google Maps API into it. From experimenting with the API it was noted that the documentation that it offers provides a well explained functions and it provides well structured code examples that anyone can access.

Use of a form was used in order to populate the input fields with the data from Google Maps API, it was reasonable challenging in order to enable to read data when the map is clicked by creating a marker on a map. Experimentation proved that marker relies on latitude and longitude data but map click returned a proper formatted address, after short research it was discovered that using additional service Geocoding, Geocoding can translate human readable address into longitude and longitude. Doing so enabled to change the clicked position on the map into the data that is required for the marker.

Prototype resulted in a success as it allowed to discover what kind of information is going to be saved into database and how it is going to be loaded on the event page to show events location, which is going to be provided by the user input. Also it was noted that implementing Google Maps API into the project will be feasible.

Furthermore, additional research to discover if there will be any interest in such a web application was made and it proved that there is a need in the market for such a website. Both Teamer and TeamSnap have very large number of users as noted on their websites. This means that people are using such a website to either track events and the availability for them. Also it was discovered that discussions with people built a very big interest in such a website.

2.6. Methodology Selection

Sporty web application is build upon couple of features. Choosing the right methodology for the project was a key to success. There are couple of standard methodologies like waterfall (Waterfall development Methodology, 2006), rapid application development (RAD, 2016) or prototyping (prototyping, 2016), Agile (Agile, 2008) was not taken into consideration because it is most suited in team environments. Waterfall methodology is a traditional one, it is easy to manage but it needs well defined requirements also it is suitable for small and well defined projects. Rapid application development methodology is focused on delivering on time and has a time centered approach. Prototyping is complex methodology it is best suitable for projects with hard to define requirements, but it works well with high end user engagement. After closely reviewing these methodologies it felt like they do not suit this project well. Decision was made to use another methodology called Iterative and incremental development (MERA, 2016) which will suit well in the project.

This methodology works around building different features and releasing them, features are planned, evaluated, implemented and tested before they are deployed. Each iteration is finished with functioning product. This means that features are going to build up the project rather than building everything at once and then releasing it, which in theory should reduce errors, bugs and problems.

Figure 2.6 shows example of the Iterative and incremental development model iteration.

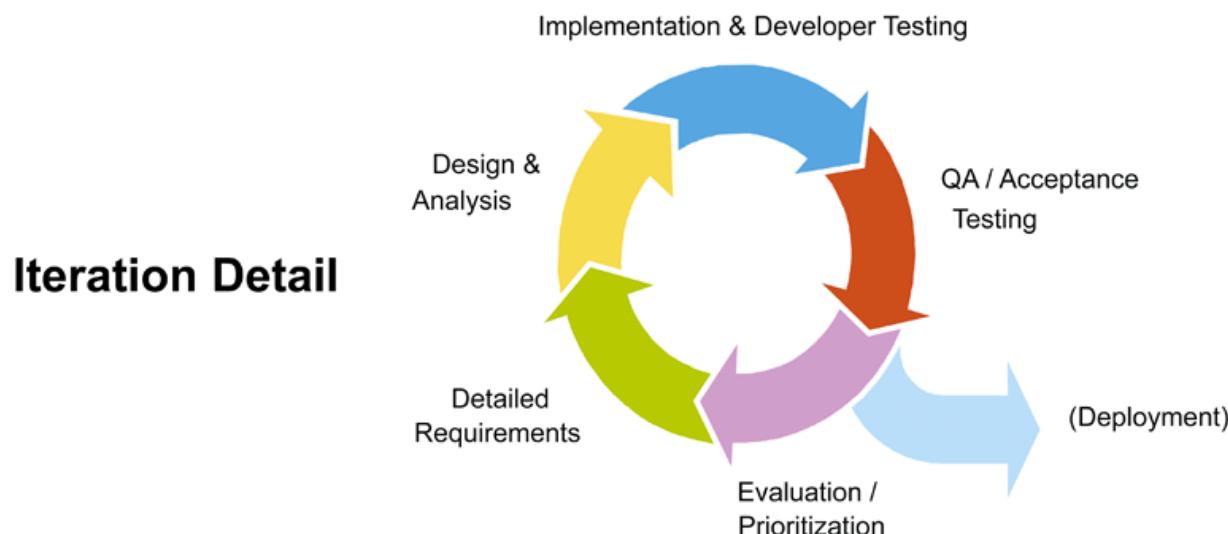


Figure 2.6 – Iterative and Incremental development methodology

3. DESIGN

Design section tackles different aspects of the design process for the application, things like User experience, system, logic and data design are covered here.

3.1. UX Design

User experience is one of the most important things when it comes to successful product, application needs to be user centred, easy to navigate and provide a simple user interface in order to create good user experience.

After creating wireframes, it was time to think about how users are going to use the site and look into their journey on the website. There was also need to design and create a logo for the website to make it look professional and give it a brand look.

3.1.1. Branding

Different ideas for the name were thought about during initial research and idea generation, see [Appendix 7]. Name ‘Sporty’ has been chosen because it felt like it is a good name for this project. It is short and easy to remember, although domain names like .co.uk or .com are already taken, thought that there is expend in different domains like .io, .uk or .me and they are becoming very popular within web industry and would be suitable for this project.

Decision was made to stick with the name and use the domains that are expending when website will go live.

Logo was created in order to give website some visual identity and representation. Following basic design process, first few sketches were drawn on paper – this can be seen in [Appendix 12]. Four of the most interesting logos were picked and chosen for further investigation. It was really important that the logo includes some sort of circle because most sports are based on a circular ball. Thought was made that this would be the essential design point of the logo. Most eye catching logo was the one at the bottom of the page, which is seen in [Appendix 12] as it gave impression that a ball is falling into some sort of playing field.

Decision was made to take this logo further and design it properly. Figure 3.1.1 shows the final design of the logo.



Figure 3.1.1 Sporty logo

Furthermore, logo design inspired to choose appropriate font for the logo and website.

Looking at the budget of the project, there was no resources to buy any fonts. It was vital to choose a free font with appropriate licence. Research was carried out to find the best suitable font for the project. Websites like Google Fonts (Google Fonts, 2016), Fonts Squirrel (Dunham, 2009) and DaFont (DaFont, 2016) where used in order to view different collections and test the fonts. Initial testing showed that the best choices are; font Archistico (Archistico, 2009) with a free commercial use this font will be used in Logos. Font Roboto (Roboto, 2016) offered by Google Fonts was chosen to be used as a main text with in the website. The choices were based on how the fonts suit the project and style of the website, different fonts were installed and tested within the system and the best choices were chosen.

Website also required a professional photo for the main page. In order to make the website professional a search for an image that would suit the project was carried out, it was vital again that the image is free with appropriate license to use. Few websites were used to find suitable image, Stacksnap (Stacksnap, 2016) that offers hundreds of high resolution images that are under Creative Commons CCO license which means images are completely free even for commercial use. iStockphoto (LP, 2016) was used to check for potential images but the website does not offer free images, however images are in very high quality. Pexels (Pexels, 2016) which works on a similar way as Stacksnap and the images are under CCO

license. It took a lot of time in order to find the image that suits this project. Image is going to be the first thing that the user is looking at when visiting the website, it was necessary that the image translated what the website is about. It was also important that the image would send a message that it is about meeting random people and joining together to play a sport. Chosen image (Parzuchowski, 2016) for the project can be seen in [Appendix 13].

Next step was to find icons which can be used within the website. Bootstrap itself includes some of theglyphicon that will be used in the project. But it was necessary to find icons that would represent each sport. Icons were found on Freepik (LLC, 2010) this website offers very high standard icons that are free, the only condition is to give a link to the website within the footer section. Each sport that is listed in the website will be represented by icon and specific colour. Doing so will make it easier to find what the user is looking for visually. Icons and the licence can be seen in [Appendix 14].

3.1.2. User Journey

In order to distinguish different users, few user personas were created using Xtensio (Nguyen, 2016). Doing so showed different groups of potential users, and allowed to capture and express their potential needs, motivations, values or traits. Figure 3.1 is one of the example of user personas, rest can be seen in [Appendix 8].



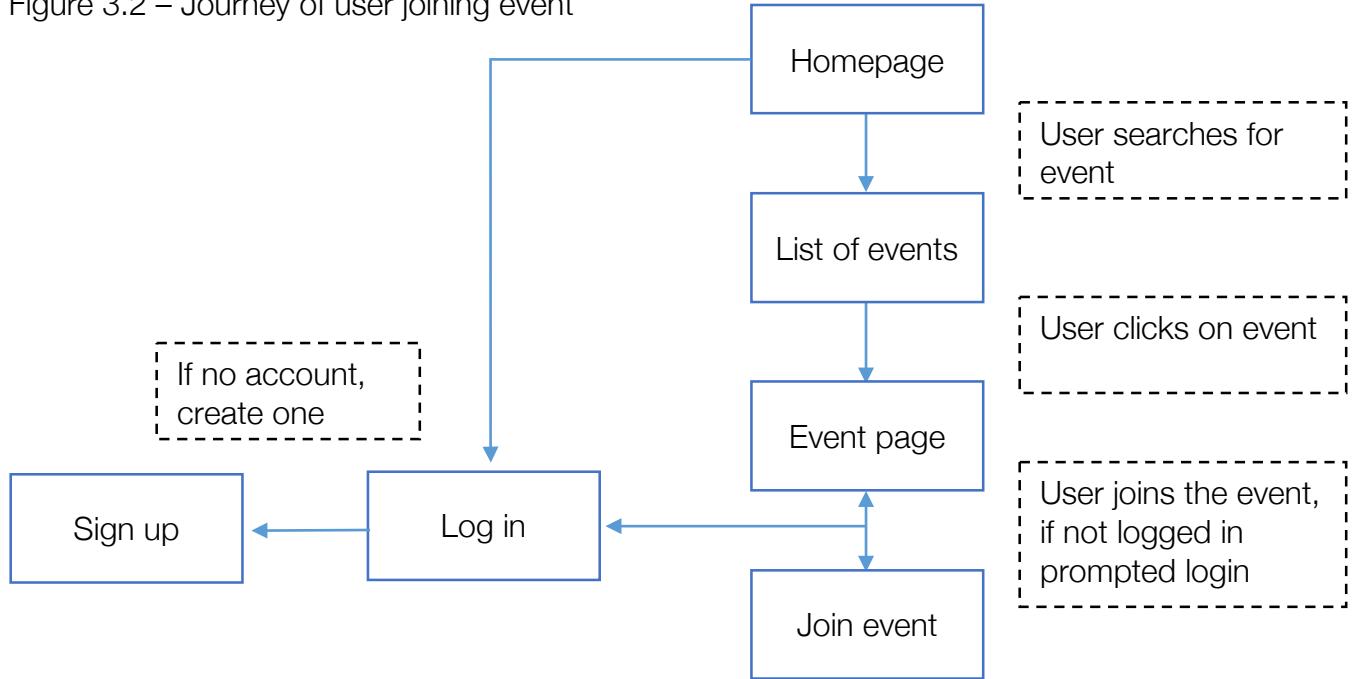
Figure 3.1 – User Persona for person looking for events

There are two main potential users that are going to use the website, they are:

- Person looking to join event
- Person looking for people to join their event

Taking a look at the first potential user, this user is looking to join event. This user will use search feature in order to find what they are looking for. Figure 3.2 shows example of this user journey on the site.

Figure 3.2 – Journey of user joining event



Next potential user has different needs. Goal of this user is to find people for the event. First thing that this user will do is create an event and publish it. Figure 3.3 shows the example of this user's journey.

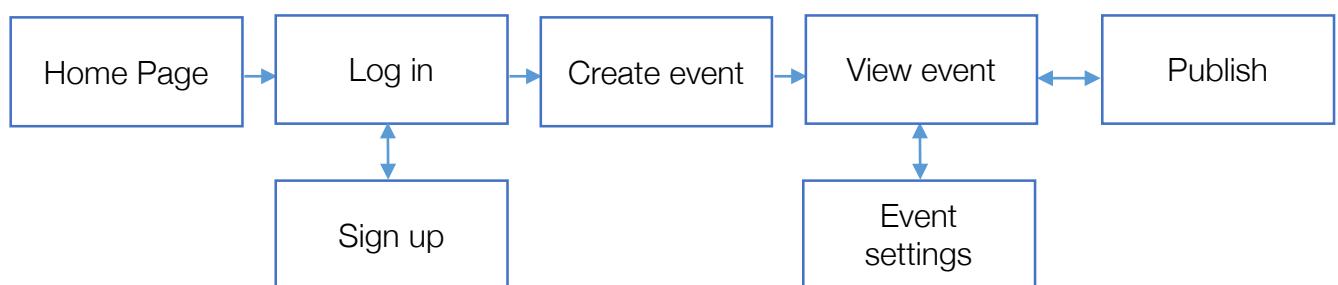


Figure 3.3 – Journey of user creating an event

Now that the journey of the two main users are established, their experience so far is going to be simple and straightforward. As you can see they are able to access what they need within few clicks. There are however two different types of the users, account holders and non-account holders. Most of the features on the site will be accessible by all of the users with no-account, but certain actions like signing up for event or creating event itself will require a user to sign up. Example of this can be seen in Figure 2.1.

3.1.3. Responsive Design

Responsive design is a crucial part of the design as more people access websites through their phones than computers (Hern, 2015). It is necessary to make the website work on mobile devices. There is huge number of different devices, from phones to tablets they all have different screen sizes, resolution and specs. Testing phase will take important part of responsive design, it will state important functionality or design issues. Google chrome offers device mode that can be used to check the website in different devices (Tools and Performance, 2015).

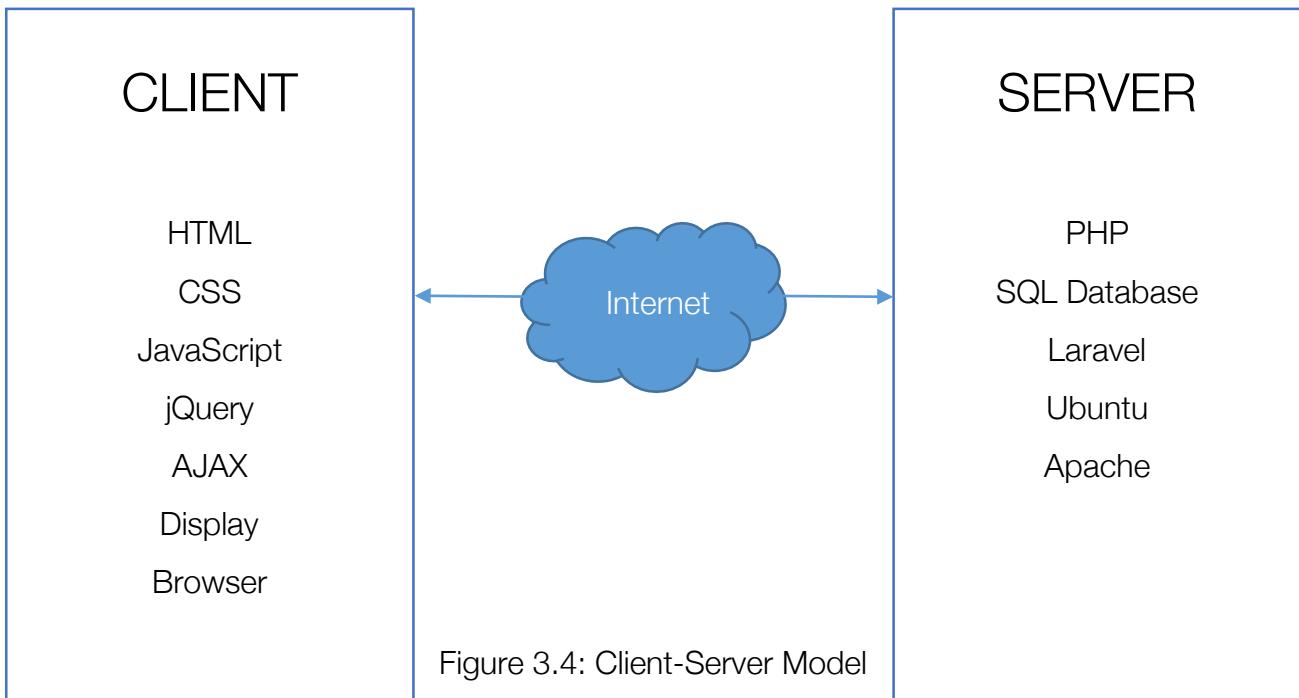
3.2. System Design

System design outlines the technologies selection and architecture of the system as well as looking into different design patterns.

3.2.1. Client-Server Model

Client-Server model describes the technologies used on the client side and the server side for the system. There are different technologies used in both server and client side, understanding how each of the programming languages functions and where it is best used is very important. Different projects and assignments were completed, to learn client and server side technologies.

Figure 3.4 shows what programming languages are being used on which side.



3.2.2. Technology selection

Client-side:

While thinking about client side technologies, it was required to think about mark-up, styling, and client-side scripting languages. Looking into the mark-up decision was made to use latest HTML version. HTML 5 has the most advanced technologies like geolocation which can be used to track users position and new semantic, graphic and multimedia elements. HTML 5 will make application up to date with latest technologies and all major browsers have support for these new features, that is why HTML 5 was chosen.

When looking into styling, decision to use SASS or LESS pre-processor to compile CSS3 needed to be made. After comparing two technologies, choice to use SASS was made because of a previously using SASS and having experience using it, it would be required to spend some time learning syntax for LESS which can be used to learn something else. Both LESS and SASS offer similar functionality like nesting and the ability to use variables.

Bootstrap framework will also be used as CSS Framework. Decision to choose Bootstrap over Foundation was made because Bootstrap is already build into Laravel, having experience using Bootstrap from different university modules as well as placement it will

again save time and allow to concentrate on building different features instead of spending time learning new CSS Framework.

As for client-side scripting languages it was very obvious to use JavaScript language, this is because this language is taught through out course as well as it is the programming language for the web. With additional to JavaScript, jQuery library will be used which simplifies JavaScript programming.

In order to make Sporty rich internet application and create better user experience by exchanging data between client and server without the need of reloading the page, AJAX will be used to support that.

Server-side:

When looking into server side technologies, there were few options to choose from. PHP, Ruby or JavaScript could be used for server side scripting, with rise of Node.js (Node, 2016) JavaScript can be used in the back-end. For this project PHP was the language of choice as most of interactive websites are build using PHP which makes it very popular language with a lot of help and resources online. PHP is simple to install on a server and it supports most databases making it solid choice for web application.

There are many different PHP Frameworks (Pty, 2015) like Symfony2, CodeIgniter or Laravel, these seem to be the most popular choices. After looking into these frameworks it was decided to use Laravel. This decision was based around few aspects, documentation of framework and help available on the site, help available online from different sources like videos and tutorials, learning curve, speed and performance and popularity. Laravel seems to be the right choice as it is valuable framework to learn for future and it provides everything needed to make the project successful.

Database:

Database is where collections of data is stored, data is manipulated with-in server and can be retrieved / updated / created or deleted with server-side language. Having used only MySQL which is most popular Open Source SQL database management system it is safe to say that the choice of database language is MySQL. There are however different options

available like SQLite or PostgreSQL. After short research (DBComparison, 2014) to find out little bit about these engines, decision was made to go with MySQL as having necessary experience using it will allow to concentrate on building different features also MySQL is a good engine for websites, it is fast and capable of handling a lot of requests.

Other:

Google Maps API will be used in order to provide a visual map of where the event is located, Geolocation will be used which will provide the geographic location of the place which will be used to track the address as well as longitude and latitude of the events.

Gulp (Gulp, 2016) is a task/build runner, it allows minification or compiling SASS files. Gulp also comes with many plugins in order to streamline build system. Laravel uses Elixir which is a Gulp extension that comes as a build in feature.

Version Control will be used to take a regular back-ups of the system and to allow myself to revert back if something bad happens. BitBucket (Bitbucket, 2016) offers a personal account with private repositories, BitBucket uses GIT and it is free to use.

Jdenticon (Pirttijärvi, 2015) library will be used to create a unique avatar for the user when they create account on the page, this will give the website additional feature and give a unique representation for each user. User will be able to update the avatar and provide image for avatar in settings.

SweetAlert (SweetAlert, 2000) library will be used to present user with well designed notifications. SweetAlert comes with different options and customization, it works well in either client side or server side.

Intervention (Intervention, 2016) which is open source PHP image handling and manipulation library. Intervention will handle the images that are uploaded by user for avatar, doing so will allow to have control over the width and height of the images and the size.

Laravel Collective HTML&Form (Engebretson, 2015) will help to create forms and use them with Laravel Framework, it will allow to use some of the functionality to save repeating code.

3.2.3. Model View Controller

Laravel framework is build using Model View Controller architecture. This model separates the application into tree different areas the model, the view and the controller. Model represents the data, View displays model data and sends user actions to controller and Controller provides model data to view also it takes user action actions.

Laravel also come with routing engine which maps urls to controller.

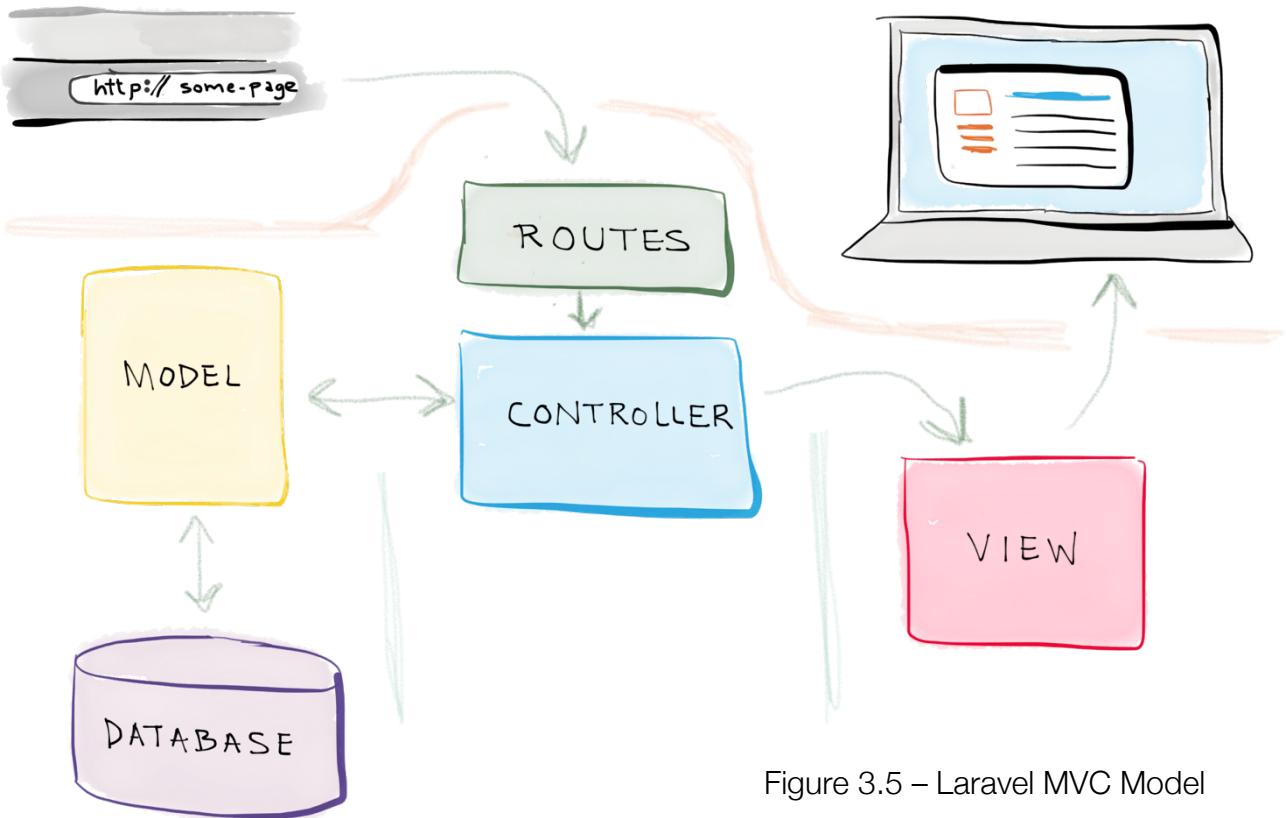


Figure 3.5 – Laravel MVC Model

Figure 3.5 shows how MVC model works in Laravel, it matches the URL to Controller which also allows variables to be send between view and controller easily. Controller handles the operations and chooses the views that are sent to the user. Model's job is to handle data, validate it and perform other operations on data which is sent to controller and views. Laravel also comes with few template languages like Twig, Smarty and Blade.

3.3. Data Design

Sporty is going to be application where most of content is created by users. Database design is crucial step as data will power the website. Entity relationship diagrams were created in order to think about different tables and how information will be stored, retrieved and

updated within the system. Figure 3.6 displays the final ER diagram; first version can be seen in [Appendix 15].

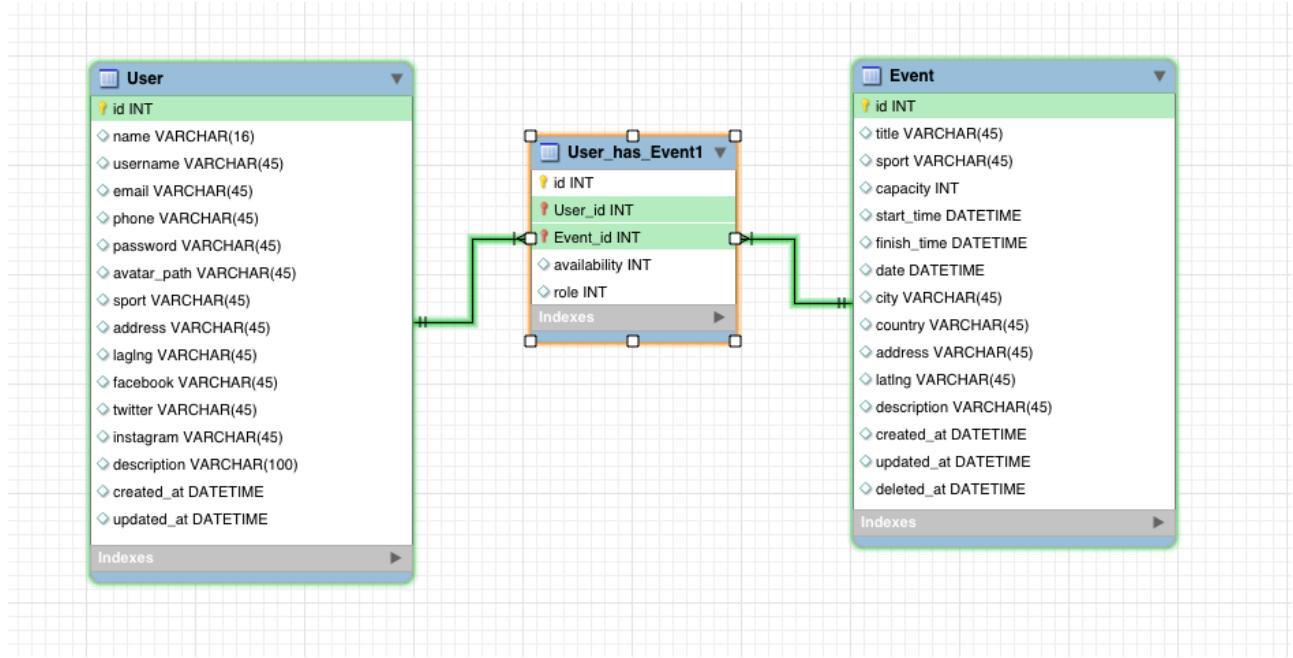


Figure 3.6 – ER diagram

ER diagram proven that there is a need for three tables to store all of the necessary data. The two main tables are Event, which holds all the data for particular event and the User table that stores all important information about users. As the relationship between User and Event is many-to-many, in between table (pivot) is necessary in order to make the data flow correctly. User is able to have many events, and the event will hold many users. The pivot table will also store the role and availability of the user for the event, the role will distinguish if the user is the creator of the event or just have joined the event to participate in.

3.4. Risks

To work out the potential risks and problems that might arise during implementation a table with risk evaluation was created. While planning the project it has raised few potential risks that could affect deadline, firstly having no experience of using Laravel 5 Framework before, it could be a big learning curve, depending on the progress of learning the framework it could affect the time in such a way that the quality or features that are planned for implementation could be affected.

Secondly having concerns about how to estimate and schedule time on specific features to develop, this could cause a huge impact on the project deadline because if project is not planned correctly it will end up not finished and the quality is probably going to be bad.

Next, being afraid that designing for cross-platform compatibility could affect the project. Need to find out how my users are going to be visiting the site, and which browsers they are using. Potential waste of time to develop and support IE8 design when none of users are using it. Checking what are the most used browsers and mobile devices could be a good idea.

Concerns about the design and branding of project were raised because it was noted that these are weaknesses and will probably need more time than needed to implement them and again the time and planning could be the key here. Possibility to overcome this can be by applying the things learned during course.

Below is a table with risks evaluation:

Risk	Risk Level	Plan	Backup
Learning Laravel	5/5	Having zero experience using Laravel, it is going to be a main risk as the plan is to build the system using this framework. In order to learn it, plan to go through guides and use Laracast website to help to learn and understand it.	Fail to learn to use the framework or Laravel turns out not to be the framework to build the system with, decision will be made to build the system using no framework.
Google Map API	4/5	One of the big parts in the system will be use of maps in order to show where the event will take place as well as using Geolocation to check users position and display events that are	Failure to integrate Google Maps into the system will surely reduce the user experience as well as it is going to be much harder to search for events and display events using Geolocation, alternative

		<p>near user. Not used Googles Map API before and feeling that it is going to be a tricky challenge to integrate. Plan is to use documentation that is provided by Google.</p>	<p>service like Bing Maps will be used if Google maps fails.</p>
Database Design	4/5	<p>Database is crucial part of the system, Plan to having a excellent Database design by firstly planning, implementing and testing the design. DB should also be scalable and extendable.</p>	<p>If it fails to design database correctly from the start and it turns out that the connections are not working correctly, it will cost valuable time to fix. Back up for this is to spend extra time to make sure it work correctly to prevent future problems.</p>
Programming Languages	2.5/5	<p>Project is using few programming languages, there will be challenges that will require some extra learning. Plan to tackle challenges by firstly trying to solve it by myself.</p>	<p>Failure to solve the coding challenge and being stuck on it, will require going to seek help from colleagues or tutors so that it can progress and finish of the feature.</p>
Version Control	1/5	<p>In order to prevent loosing the project files, plan is to use version control system. This will keep a regular copy of the files and enable to access it from different devices, not only personal computer.</p>	<p>If for any reason version control system is down, plan to also take a regular copies of the files and save them to external drive.</p>

4. IMPLEMENTATION

This section reflects on the projects implementation. Different technologies were used during the project, implementation chapter looks into technology review, technology use, notable challenges and notable achievements.

4.1. Technology Review

From the project planning it was decided that different technologies are going to be selected and used in the project. This section will reflect and review these technologies.

4.1.1. Client-side

Use of client-side technologies helped to enhance user experience and the website in many different ways. Project has benefited with the use of HTML 5 as new tags and features that come with it, helped to create simple but enhancing look for user profiles. Also it allowed to create a simpler to understand code structure that is easier to maintain and extend.

SASS was used as a CSS pre-processor; it was beneficial to use this technology as it allowed to create CSS that is needed for the website with much less code and it uses some programming logic to reuse the code. SASS was a good choice over LESS as it allowed to save precious time that was needed to create all the features in the project. Although some research was required to be carried out during development about few different aspects of SASS and the uses of it (Catlin, 2006).

Another client-side technology that was selected was Bootstrap Framework, this is by far the technology that the project has benefited greatly on. Not only did it save time to style all the pages within the system it has helped to create layouts for different devices, it allowed to create the website style with minimum use of CSS and enhance the user experience by quickly using its features. Project also benefits of using Bootstrap Framework because of glyphicons that it offers (Glyphicons, 2016).

The chosen programming language for client-side was JavaScript, by reviewing the use of this language it is safe to say that it has helped to integrate some of the key features within the website. Some of main libraries that are used in the project are written in JavaScript, integrating these libraries was fairly simple process and JavaScript has helped to either use the libraries or change some of the functions to suit project needs. Use of jQuery was beneficial as it allowed to enhance users experience on the website, it made it possible to create rich application by using AJAX technology.

4.1.2. Server-side

Laravel framework was the chosen back-end technology. One of the main risks was the use of this framework as it involved a big learning curve. Learning Laravel was very time consuming but with the help of the videos from Laracasts (Laracasts, 2016) it was possible to learn much quicker and much more effectively. Only some of the videos from Laracasts are free, and it was beneficial to pay for subscription to speed up the learning. Project eliminated the main risk and framework was the perfect choice for this application.

Laravel itself turned to be very solid choice for the project, it allowed the use of few development tools that made implementation of the website smoother and cleaner. Setting up local development with the help of virtual environment allowed to create a server like area where the code was tested and behaved as it was on production server. Laravel uses some of well known technologies within web industry like node.js and composer (Composer, 2016) to maintain all the different packages, which makes it very simple and easy to install additional packages.

Framework also comes with Elixir, which is a task runner to enhance workflow. Elixir has helped to speed up some simple tasks and made developing the application better experience by reloading the pages automatically on changes, or taking care of minifying and uglifying script files for production.

Overall Laravel proved to be a great choice and made both the application and development tools a great benefit. Laravel is build on PHP language, using PHP allowed to create a

website that can be run on many servers and is very scalable. Language itself allowed to write maintainable and scalable code.

Database turned to work very effectively as it was planned and tested well before implementation, no major changes were noted and it was a relief as one of the main risks was eliminated. Laravel comes with Eloquent ORM (Eloquent, 2016), which made working with Database simple. Additional benefit of Laravel is Database: Migrations (Migrations, 2016), which helped to maintain a database like a version control so it was easy to modify and change the scheme of the tables.

4.1.3. Other

Other technology that helped to implement the website was mentioned in planning phase, different libraries and tools used helped to implement some features quicker and with much nicer result.

4.1. Technology Use

Many different tools and technologies were used to create Sporty, this section will highlight the use of these tools and technologies.

While reviewing the client-side technology it was stated that HTML5 has helped to enhance user profiles. Use of HTML5 can be seen in creation of navigation where new tags were used, another example of HTML5 is the use of canvas technology (Canvas, 2016) where user avatar was created when registering. Jdenticon library has helped to integrate creation of unique avatars, that are based on user's username.

The use of Bootstrap framework allowed to create layout for all the pages and maintain high standard design, bootstrap helped to develop quick solutions to layout problems. Nearly all of the Bootstrap features were used in the project; Account settings page uses a lot of bootstrap features like Pills, that enable to create account settings navigation. Tooltips, to display label text for all the different options under events, Modals to display event options when clicked as well as the use of glyphicons, tables, forms and Panels. Overall project has

used a lot of features offered by Bootstrap, which creates simple, yet enhancing user experience.

With the use of JavaScript and jQuery it was possible to create a visual enhancement when the user decides to join the event. Event page includes a simple progress bar that displays the number of users that joined the event, by clicked the join button AJAX call is made and user is added to the event in the system, this results in displaying a successful message with the use of SweetAlert library, making the progress bar slide as well as unveiling additional settings for joined members, user is then able to see rest of members for the event as well as change status.

Server-side technology resulted in use of many different tools. In order to create local development, use of vagrant (Vagrant, 2016) and virtual box (VM, 2016) with homestead (Homestead, 2016) was used to create server like area. This was the first step before starting any development, Installation of Laravel framework was necessary but following the documentation it turned to be simple process.

Another tool that was used in the project was the use of version control, which has helped to maintain some solid backups. Bitbucket was the chosen Git version control. By pushing the code to the repository, it was possible to quickly start working on different computers by simple pulling the repository to the local machine.

4.3. Challenges

During development there were few challenges that either were already noted in the planning phase and there were few challenges that were raised during development of the project. This section will dive into the implementation of the challenges and the solutions that were created for them.

Google Map API:

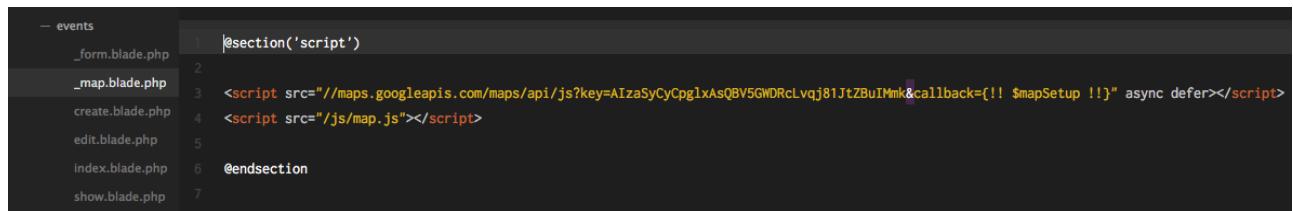
Google Map API was noted in the planning phase as one of the main risks. Soon after the creation of first prototype to check the feasibility, which resulted in minimizing the risk as the

integration of prototype resulted in success. However, there were still some challenges involved on the way when integrating the Map in the system.

The logic behind the map and what was required, was to use Google Map API to enable the user when creating the event to select the location of the event by clicking on the map or by simple typing in the address in the search bar to find the address. Event page would then show the location with marker but user or event creator would not be able to change the markers location on that page, only by changing the event settings by event creator address would be changeable or other event settings.

Later on it was discovered that the functionality can be extended by adding an option where user is able to select their own location and appear on the event page if they have joined the event so that other members are able to see the location of each other, this feature would potentially help users to find lifts and manage drivers needed for the event.

As the map appears on more than one page it was important to minimize the code and refactor it in a way where the code can be reused across the pages. Within Laravel a view was created that calls in the appropriate function needed for the page and it is responsible for loading map file. Figure 4.3.1 shows mapSetup parameter that is passed to the view where it is included.



```

-- events
  _form.blade.php
  _map.blade.php
  create.blade.php
  edit.blade.php
  index.blade.php
  show.blade.php

1 @section('script')
2
3 <script src="//maps.googleapis.com/maps/api/js?key=AIzaSyCyCpgIxAsQBV5GWRcIvqj81JtZBuIMmk&callback="{!! $mapSetup !!}" async defer></script>
4 <script src="/js/map.js"></script>
5
6 @endsection
7

```

Fig 4.3.1. Google Map view

This has helped to manage the calls to the scripts and only load them when needed and place all the functions for the map inside one JavaScript file. The technical challenge was to get all the necessary data from Google Maps API when user creates the event. When user wants to search for the event, they are advised to search for location. Which can be either city or country, unfortunately Google does not make it easy to retrieve this kind of information as it creates full address from small pieces and a specific function needed to be created in

order to retrieve just city and country of the clicked place by user. With the help of Google documentation, it was possible to understand the retrieved object from API and get city and country. Figure 4.3.2. shows main part of the function.

```

138     $.each(val['address_components'], function(index, value) {
139         if ($.inArray('locality', value['types']) == 0){
140             city = value['long_name'];
141         } else if ($.inArray('country', value['types']) == 0) {
142             country = value['long_name'];
143         } else if ($.inArray('postal_town', value['types']) == 0) {
144             postal = value['long_name'];
145         }

```

Fig 4.3.2. Part of findCityCountry() function

The rest of the data needed to create event was tested in prototype and was used in the main code, Address, Latitude and Longitude was retrieved with the help of Geolocation from Google Maps. All of this information is saved to hidden form fields and then saved to Database on creation. Similar solution was integrated on saving user's location.

When retrieving the information and initializing the map with the location on the event page, function showMap() was created. The technical challenge on this page was to not only get the location of the event but also get all the members that have set their location in settings to display on the map. With the help of jQuery simple logic was implemented that enabled to loop through each of the users that have set their location and add them as markers to the map. Figure 4.3.3. displays the function to add markers for the users.

```

187     function createUserMarker() {
188         $('[data-location]').each(function() {
189
190             //if no location provided by user ignore and continue
191             if ($(this).attr('data-location') == 'noData') {
192                 return true;
193             }
194

```

Fig 4.3.3. Part of createUserMarker() Function

With addition to functionality some of the design was added like custom marker for events, simple drop animation and default custom avatar picture for user's marker, or if user has set up their account with profile avatar picture then it is used. Choice of using Iterative and

incremental development methodology helped to develop Google Map API functionality fully before moving to another feature in the project. Figure 4.3.4 displays map on event page.

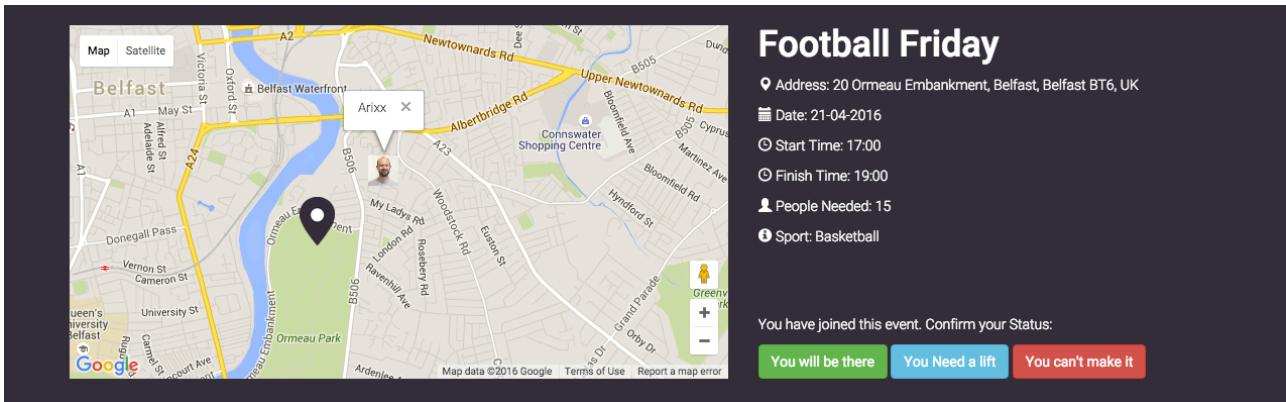


Figure 4.3.4. Event Page Google Map Example

Laravel:

Laravel Framework was noted as another of the main risks, but with the help of online resources potential risk of learning the framework was minimized and reduced to minimum. Sporty is build upon Laravel 5.2, it is the heart of the application as it tells it what to do.

At first it was a big challenge to learn all the different aspects of the system and how it works. Laravel uses command line interface to help with common actions, Artisan (Artisan, 2016) is the name of the CLI that comes with Laravel. It comes with very useful and helpful commands that speed up the development process. Figure 4.3.5 Displays use of artisan CLI

```
→ major_project git:(master) php artisan list
Laravel Framework version 5.2.29

Usage:
  command [options] [arguments]

Options:
  -h, --help           Display this help message
  -q, --quiet          Do not output any message
  -V, --version         Display this application version
  --ansi               Force ANSI output
  --no-ansi             Disable ANSI output
  -n, --no-interaction  Do not ask any interactive question
  --env[=ENV]            The environment the command should run under.
  -v|vv|vvv, --verbose   Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  clear-compiled        Remove the compiled class file
  down                  Put the application into maintenance mode
  env                   Display the current framework environment
  help                  Displays help for a command
```

Fig 4.3.5 Use of Artisan CLI to display all the available commands.

Routing and middleware was used to create the applications and specify what should be displayed to the user based on URI, doing so allowed to distinguish dynamic parameters that and static pages and specify which controller should be used with the accessed URI. The key with routing was to state the static routes before the dynamic ones as it would then treat static routes as dynamic. Sporty follows RESTful routing. Figure 4.3.6 shows Sporty routes.



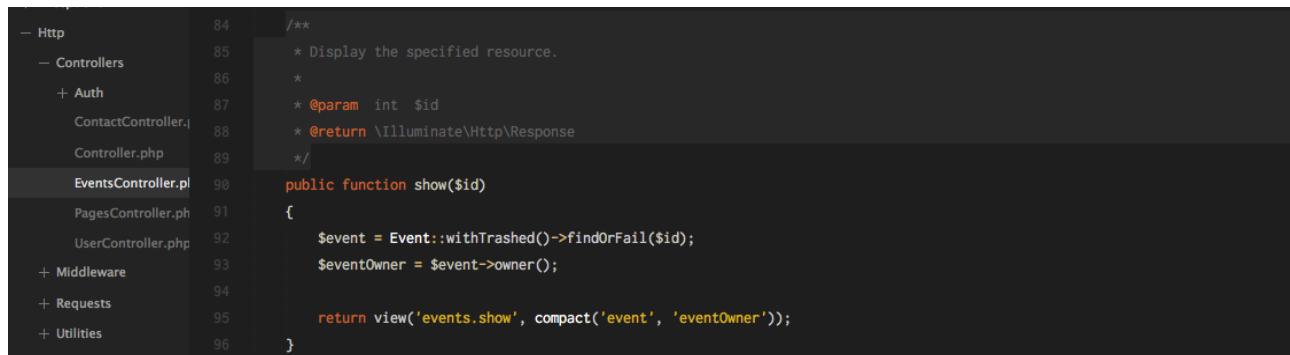
```

12     Route::group(['middleware' => 'web'], function () {
13         Route::auth();
14
15         Route::get('/', 'PagesController@index');
16
17         Route::get('contact', 'ContactController@index');
18         Route::post('contact', ['as' => 'contact_send', 'uses' => 'ContactController@send']);
19
20         Route::post('events/{id}/join', 'EventsController@join');
21         Route::post('events/{id}/status', 'EventsController@status');
22         Route::put('events/{eventid}/users/{userid}', 'EventsController@adminchangestatus');
23         Route::delete('events/{eventid}/users/{userid}', 'EventsController@adminremoveuser');
24
25         Route::resource('events', 'EventsController');
26
27         Route::get('settings', 'UserController@edit');
28         Route::get('{username}', 'UserController@show');
29         Route::put('users/{id}', 'UserController@update');
30         Route::put('users/{id}/newpassword', 'UserController@updatePassword');
31     });

```

Fig 4.3.6 Sporty Routes

Every route is linked to a controller; controller holds all the logic that should be done with the data and what view should be displayed to the user. Controllers are easily created with Artisan command; artisan make:controller. Sporty has two main Controllers to handle the logic behind events and users. These controllers include different methods where URI calls are directed to, when user is viewing a particular event, following the RESTful routing it sends the route to show() method on event controller where it deals with all the logic for this request. Figure 4.3.7 shows the show method in events controller.



```

84     /**
85      * Display the specified resource.
86      *
87      * @param int $id
88      * @return \Illuminate\Http\Response
89     */
90    public function show($id)
91    {
92        $event = Event::withTrashed()->findOrFail($id);
93        $eventOwner = $event->owner();
94
95        return view('events.show', compact('event', 'eventOwner'));
96    }

```

Fig 4.3.7 Show method in Events Controller

Controllers also enable to state to which methods user needs to login in order to access them, for example in order to access settings – user is required to login first. This is possible because of using middleware and is specified in `__construct()` method in controller. The use of different models is used in the controllers, which handle different database requests.

Application consists of two main models where it is specified which fields are fillable, website also required the use of Carbon (Carbon, 2016) which allowed to format the time and date when the request is made to get them from database. Figure 4.3.8. Displays use of Carbon.



```

+ Providers
  Event.php
    Functions.php
    User.php
+ bootstrap
+ config
+ database
+ node_modules
+ public
- resources
  63
  64
  65
  66
  67
  68
  69
  70
  71
  72
  73
  74
} */
 /**
 * Get events finish time.
 *
 * @param string $value
 * @return string
 */
public function getFinishTimeAttribute($value)
{
    return Carbon::parse($value)->format('H:i');
}

```

Fig 4.3.8. Use of Carbon in Event Model to change format of finished time

Controllers also decide what view is returned to the user, view represents the visual representation of the page that is displayed on the URI call. Blade (Blade, 2016) was used through out all the views as it enabled to manipulate the data and use some of the programming logic like loops, if statements and use Laravel Collection (Collections, 2016) methods.

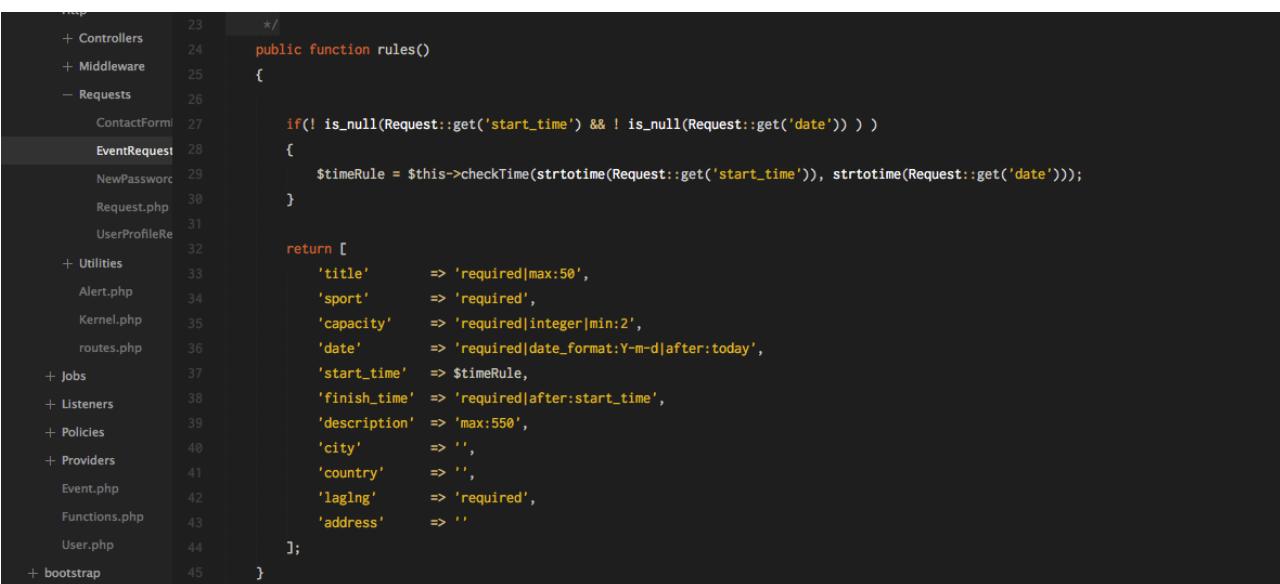
Laravel created a lot of challenges on the way of learning and in development stage, but with the determination, Laracasts and use of documentation it was possible to build the website. Laravel offers a lot of functionality and features, the use of migrations helped to achieve a version control for database tables, artisan commands helped to maintain quicker development and easier integration of new feature to the system.

Forms:

There are few forms used within the system, some of them proved with challenging aspects especially when it comes to validation. Few of the forms where integrated with the help of Laravel Collective Forms & HTML, which made it possible to reuse the same code for creating and editing events as it allowed to use model binding and create well functional input

fields like data field in which user is able to select calendar and pick a date, this form can be seen in _form.blade.php file. Laravel also protects from cross site request forgery (CSRF, 2015) by creating a hidden input field with the csrf_token. Validation was one of the challenging aspects for the forms, especially when it came to creating events.

There was need to put a logic in place where the user would not be able to create events in the past. Sporty has three different input fields that deal with validating the date and time, for example user should not be able to create events in the past so when user selects date and it is in the past, error message should be prompted. When user select todays date but chooses the start time in the past, again error message should be displayed. When user decides to choose proper start time and date, but selects the finishing time that is before start time, again error message should be displayed. Validation should pass when the date, start time and finish time is correct, finish time needs to be greater than start time. In order to achieve validation for the forms, the use of Requests was made where it was possible to store all the validation logic for different forms. This allowed to separate controller logic and validation. Figure 4.3.9. Shows example of Request file with validation for event form.



```

    +-- Requests
      +-- EventRequest.php
  
```

```

  23  */
  24  public function rules()
  25  {
  26
  27      if(! is_null(Request::get('start_time')) && ! is_null(Request::get('date')) )
  28      {
  29          $timeRule = $this->checkTime(strtotime(Request::get('start_time')), strtotime(Request::get('date')));
  30      }
  31
  32      return [
  33          'title'      => 'required|max:50',
  34          'sport'       => 'required',
  35          'capacity'   => 'required|integer|min:2',
  36          'date'        => 'required|date_format:Y-m-d|after:today',
  37          'start_time'  => $timeRule,
  38          'finish_time' => 'required|after:start_time',
  39          'description' => 'max:550',
  40          'city'        => '',
  41          'country'    => '',
  42          'latlng'      => 'required',
  43          'address'     => ''
  44      ];
  45  }
  
```

Fig 4.3.9 Shows validation rules for creating event

Laravel comes with many different validation rules (Validation Rules, 2016) but in order to validate the date, start time and finish time, custom validation rule needed to be created. Simple method was implemented called checkTime() that would check the logic and return the validation rules that are needed.

Event Cycle:

One of the challenging aspects when it came to event life cycle was thinking about how to handle the problem with when the event actually starts, it would not be very efficient to manually soft delete the events as this would require a developer or a person with specific permissions to constantly check all events and their start time, even then it would be very likely that the person would miss some events. System needs to be responsible for checking when the event has started and soft delete it. Soft delete is used because there is a need to show the events that have been completed in user's settings, profile and it should be viewable.

Integrating this piece of functionality resulted in some major changes that needed to be taken into consideration, starting with the list of the events that the user is searching for. Straight away there was need to check if results include soft deleted events, Laravel Eloquent is smart enough not to include soft deletes in the queries when not needed. It is possible to include them with special method `withTrashed()` that includes the soft deletes in query. Figure 4.3.10 Shows the use of `withTrashed()`.

```
public function show($id)
{
    $event = Event::withTrashed()->findOrFail($id);
    $eventOwner = $event->owner();

    return view('events.show', compact('event', 'eventOwner'));
}
```

Fig 4.3.10. Example of `withTrashed()` method in events controller

Implementing the logic behind who can access and view the different features on the events was challenging task. There are different stages in which the user interacts with specific event page. Looking at the user who is not registered, the options that are displayed for this type of user is just a login button, because to join the event user needs to be logged in. If user decides to access event that is soft deleted through URI, message displayed should be that the event has been finished.

When user is logged in there are two different scenarios, user who has joined the event and user that has not. User who has not joined the event should see join button, if accessing soft

deleted event through URI it should display that the event has finished, same as with the user that is not logged in. While the joined in user should see the full options, however user should not be able to update status if the event is soft deleted.

It was tricky and challenging to implement all of the different stages, mainly all of this logic was implemented in the view because it was manipulation of the data received from controller. Figure 4.3.11. Shows the logic behind showing different user options.

```

47 <div id="user-option" class="col-md-5">
48     @if (Auth::guest() && $event->deleted_at == null)
49         <a href="/login"><button type="button" id="event-login" class="btn btn-primary">Login to join</button></a>
50     @elseif (Auth::guest() && $event->deleted_at != null)
51         <p class="btn btn-danger">Event Finished</p>
52     @elseif ($event->users->where('id', Auth::user()->id) != [])
53         <button id="members-show" type="button" class="btn btn-warning" value="members">Event Members</button>
54     @elseif ($event->deleted_at == null)
55         {!! Form::open(array('action' => array('EventsController@join', $event->id), 'id' => 'joinevent_form')) !!}
56         <div class="form-group">
57             {!! Form::submit('Join Event', array('class' => 'btn btn-primary')) !!}
58         </div>
59     {!! Form::close() !!}
60     @endif
61 </div>

```

Fig 4.3.11 User-option logic from event page, view file show.blade.php

As mentioned above the system needs to check for events that have started and soft delete them. In order to achieve this a specific CLI command was implemented to use with artisan, which is executed with Cron job that runs every minute to check for events in the background. Decided that the command will be executed by sport:check-events, also there was need to enable the command in the Kernel.php file so that it is provided with artisan commands. With the use of carbon it was possible to write a logic inside handle() method, which can be seen in below figure 4.3.12.

```

40     public function handle()
41     {
42         $now = Carbon::now('Europe/London');
43         $events = Event::whereDate('date', '<=', $now->toDateString())->get();
44
45         foreach ($events as $event) {
46             if ($event->start_time <= $now->toTimeString()) {
47                 $event->delete();
48
49                 echo "Event: " . $event->id . " -- Deleted at:" . $now . " -- Start time was: " . $event->start_time;
50             }
51         }
52     }

```

4.4. Achievements

During the implementation of the website there were few features and tools that stood up, this section highlights different achievements for building the website.

Profile Picture:

Within the system users have their avatar created when they register, but they are able to change it in the account settings. Implementation of this feature required the use of Intervention library that helped with resizing and saving the image to file system.

There were few aspects when developing this feature, first with the file input field it was simple to enable to let the user choose the file from user's computer. But doing so enabled to choose all the files possible from the file system, adding 'accept' attribute to the input field with the value 'images/*' allowed only images to be picked. Back end validation was also implemented in UserProfileRequest file. Decision was made to spend some time to enable the image that is selected by the user automatically viewable in the avatar field to enhance users experience. This was achieved by creating a readURL() function in the main JavaScript file, this function uses FileReader() to load the selected image. With the help of jQuery it was possible to listen for the change on this avatar field and call the function to replace the image with the new one. Nothing is send to the server at this point. Only by clicking the save button, information is send to the server and saved to database and file system.

When the call is made and the validation Request has passed, update() method in UserController is called to deal with the data. Picture name is then hashed with time, to avoid possibility of overriding pictures with same name as shown in Figure 4.4.1

```

64     if ($avatar) {
65         $name = sha1(
66             time() . $avatar->getClientOriginalName()
67         );
68
69         $fileName = $name . '.' . $avatar->getClientOriginalExtension();
70         $filePath = 'images/photos/' . $fileName;
71
72         Image::make($avatar)->fit(150)->save($filePath);
73     }

```

Figure 4.4.1 Image part of update() method in User Controller

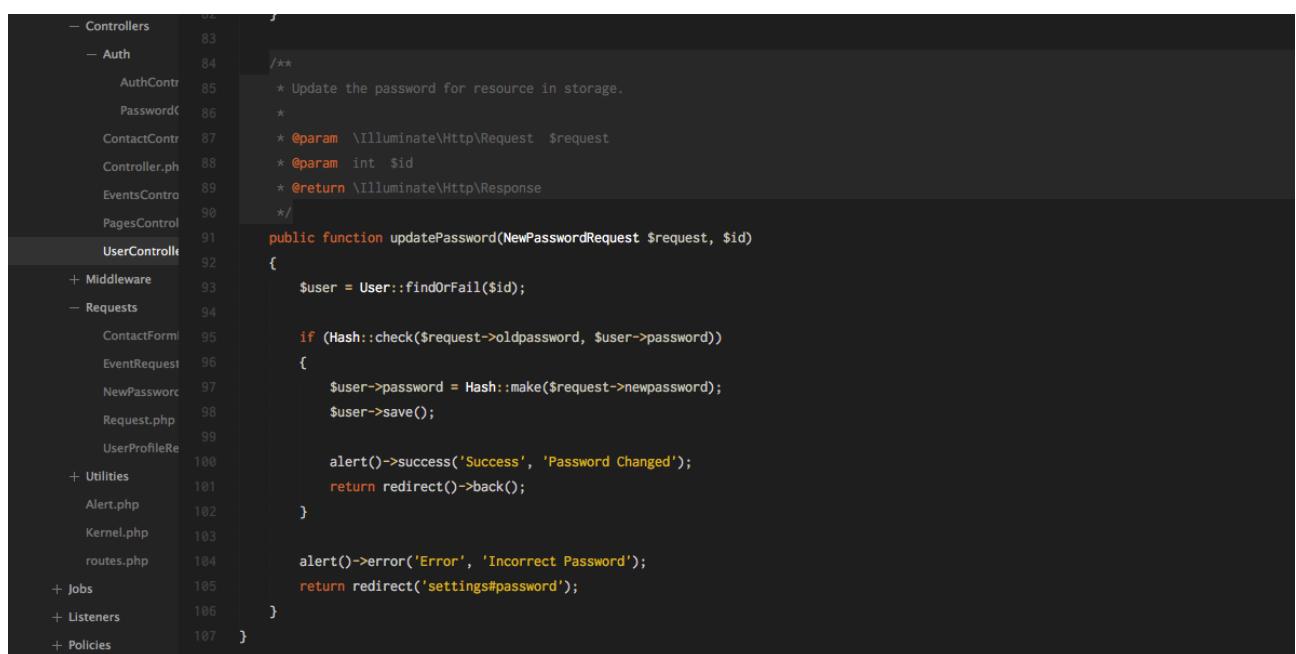
Login / Signup:

Laravel 5.2 comes with build in authentication layer, this made is very easy to implement login and registration. Running one artisan command installed all necessary scripts that enabled the user to; register, login and forgotten password (although email needs to be setup separately in order to send details).

It was then very easy to add username field to the registration form and validation for it, Laravel proved to be the right choice once again as it allowed to save valuable time to concentrate on implementing different features.

Password Reset:

Unfortunately, Laravel does not include a password reset functionality in authentication layer, and this feature needed to be implemented. Users are able to reset the password in account settings. This feature provides a simple form where user needs to insert their current password, new password and confirmation for new password. Another Request file was created to handle the validation and the logic was implemented inside UserController, where it checks if the old password matches the current password and then updates the password field. Figure 4.4.2 shows the code to update users password.



```

    /**
     * Update the password for resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function updatePassword(NewPasswordRequest $request, $id)
    {
        $user = User::findOrFail($id);

        if (Hash::check($request->oldpassword, $user->password))
        {
            $user->password = Hash::make($request->newpassword);
            $user->save();

            alert()->success('Success', 'Password Changed');
            return redirect()->back();
        }

        alert()->error('Error', 'Incorrect Password');
        return redirect('settings#password');
    }
}

```

Figure 4.4.2 Method to update password

Mail:

In order to enable forgotten password to send emails with the required steps to reset user's password, a Mail server needed to be set up. Decision was made to use Mailgun (Rackspace, 2016) services as it offers 10,000 free emails every month, it allows to track every email and check if it reached the user. Another option was Mandrill (Mandrill, 2016) but it didn't offer any free options, and with the project not having any financial resources it was best to use Mailgun. Implementing Mailgun into Laravel was straight forward as Laravel comes with the needed driver, following configuration guide in the documentation. Mail was up and running, this can be seen in Figure 4.4.3.

The screenshot shows the Mailgun Logs interface. At the top, there are navigation links: Domains, Mailing Lists, **Logs**, Routes, Tracking, Suppressions, and Webhooks. Below the header, the word "Logs" is displayed in a large font. On the left side, there is a sidebar with a dropdown menu set to "sandbox3320b8fc17e...", a "LOG LEVEL" section with checkboxes for Info (checked), Warn, and Error, and an "EVENT TYPE" section with a checkbox for Accepted (checked). The main area contains a table with two rows of log entries. The columns are "Date/Time" and "Summary". The first entry is "Delivered: postmaster@ sandbox3320b8fc17eb43e08a8a22f2f6143a2d.mailgun.org → piecha-a@email.ulster.ac.uk 'Sporty - Contact Form Email'" at 04/20/16 09:30 PM. The second entry is "Accepted: postmaster@ sandbox3320b8fc17eb43e08a8a22f2f6143a2d.mailgun.org → piecha-a@email.ulster.ac.uk 'Sporty - Contact Form Email'" at the same time. There are "Refresh" and "Search logs" buttons at the top right of the main area, and navigation arrows at the bottom.

Fig 4.4.3 Mailgun logs, displaying send emails from contact form.

Simple contact form was created to test Mailgun, users are able to access the contact form in the footer section of the website.

Digital Ocean / Forge:

Sporty required a hosting server where people will be able to access the website and use it. Quick research was carried out before choosing the server provider, there were few different choices to go for; Amazon Web Services (AWS, 2016), Digital Ocean (DigitalOcean, 2016) and Rackspace (Rackspace, 2016). Decision was made to use Digital Ocean as it provided information about payments straight away, whereas the other services didn't state the pricing as it depends on the usage. Financial resources where spent on Digital Ocean subscription, it was decided to go for the cheapest option.

After sorting server provider, it was a beneficial to subscribe to Forge (Forge, 2016). Forge helps to provision and manage cloud servers easily, mainly through interface. It helps to install all the necessary requirements on the server in order to make Laravel application run. Forge is also a paid service and financial resources was provided for it.

Connecting Digital Ocean to Forge account allowed to initialize the server within Forge. Creating server was straight forward as it needed to select certain options and the deployment of the server was in progress, after short while server was up and running as shown on Figure 4.4.4.

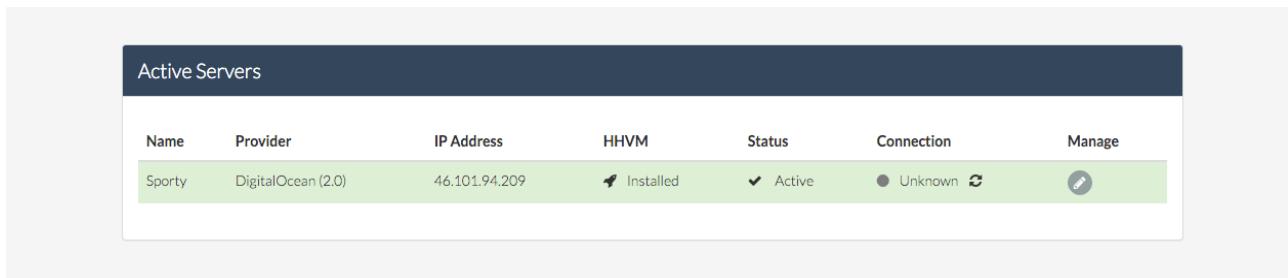


Fig 4.4.4 Active Server from Forge

Forge also allows to connect BitBucket and use feature called push to Deploy, which basically links with the repository and when there are any commits to Master branch changes are automatically deployed to the server. This is very handy as it is possible to simply push new code to BitBucket and it will automatically release changes on the server, of course some Git Flow work flow will need to be followed when the website is fully released.

Forge also provides the chance to update environmental file where it was possible to save all the environmental variables, by adding SSH key to server's website it was possible to SSH into it using CLI and run necessary commands to install database tables. Figure 4.4.5 Shows connecting to the server with SSH.

```
➔ major_project git:(master) ssh forge@46.101.94.209
Welcome to Ubuntu 14.04.4 LTS (GNU/Linux 3.13.0-79-generic x86_64)

 * Documentation: https://help.ubuntu.com/
 
 System information as of Wed Apr 20 22:04:12 BST 2016

 System load: 0.0          Processes:      97
 Usage of /: 15.3% of 19.56GB Users logged in: 0
 Memory usage: 68%
 Swap usage: 0%           IP address for eth0: 46.101.94.209
                           IP address for eth1: 10.131.7.120

 Graph this data and manage this system at:
   https://landscape.canonical.com/
```

5. TESTING

Testing is important stage and this section explains different approaches in order to test the system and all the features.

5.1. Test Approach Selection

There are few different methods that can be used to test the application, by choosing Iterative and incremental methodology most of the features are build with basic testing already done by developer and the application should work with minimum number of bugs. However, after implementing all the features it was important to test whole application and see if all the features are working correctly together.

The decision was made to use System Testing (System testing, 2016) and then a specific method called Beta Testing (Testing, 2013). Doing so will allow to check the system as a whole and compare all the functional and non-functional requirements that were planned with all the features that were implemented. Beta testing method is done by end-users and will determine if the application is ready for release.

5.2. Test Process

Table with all the functional and non-function requirements was created before running the system test. Additionally, browser testing was done that enabled to see if the application works correctly on all the major browsers. Test Process allowed to check if the system was designed and developed to allow users do what was initially planned.

Test began by simply going through the table and making sure that the application is able to do what was planned. During development some additional features were created that were not planned for. These feature were added to the table and listed in the Additional Features section. Additional features were taken into consideration because they allowed to create better user experience and provide better functionality for the system, help from mentor allowed to make justified decisions on what to develop and implement.

5.3. Test Results

The list of all the test results can be seen in the figure 5.3.1.

ID	Requirement	Outcome
#1	Allow users to register and create account in web application	Test Passed
#2	Allow users to login to the system	Test Passed
#3	Allow users to reset / retrieve password for their account	Test Passed
#4	Allow users to delete account from the system	Not Implemented
#5	Allow users to logout from the system	Test Passed
#6	Allow users to create sport events	Test Passed
#7	Allow to search for events with filter options	Test Passed
#8	Allow users to sign up for events	Test Passed
#9	Allow users to create a profile page	Test Passed – user is able to customize their settings in account settings
#10	Allow users to search for other users	Test Passed – user is able to search if they know other users username through URI
#11	System will store event information about the users	Test Passed
#12	Make web application responsive	Tested – Few issues found.
#13	System will allow event creator to cancel the event	Test Passed
#14	System will allow event creator to change details about the event	Test Passed
#15	System will allow event creator to contact all signed up users	Not implemented – however user can see members contact details
#16	System will allow signed up user to drop/resign from event	Tested – user is able to change status, and be removed by event creator

Fig 5.3.1 Test Result for requirements.

Following the system test it was noted that most of the requirements were implemented with success. Two of the requirements were not implemented #4 and #15, allowing user to delete their account from the system should not be a very time consuming feature and should be implemented if time will allow it. Contacting all the users in the system is a bit more

complicated feature, user notification system would need to be implemented in order to achieve this. Currently there is no time available for the implementation of this feature and should be noted for future improvements.

Some bugs were found while testing the system on different devices, mainly phones proved to cause small issues. Non of the issues were technical, small design issues were found and are listed as cosmetic issues that should be looked at if the time will allow it.

Additional Features that were implemented are listed in the figure 5.3.2.

ID	Additional Feature	Outcome
#1	Allow users to provide their location, and display it on the event map.	Test Passed
#2	Allow event creator to change status of joined users for particular event	Test Passed
#3	Allow users to provide links to their social media profiles	Test Passed
#4	Allow users to change statuses for events they joined and provide information if they need lift	Test Passed

Fig 5.3.2 Table of Additional features that were implemented.

Testing the additional features resulted in having a better user experience on the application. Browser testing was followed short after system testing and results can be seen in Figure 5.3.3. Decision as made to do testing on major browsers only.

Browser	Outcome
Chrome	Application working as expected
Firefox	Application working as expected
Safari	Application working as expected

Fig 5.3.3. Table of test results from browser testing

Application behaved the same across different browsers, no issues were found during the testing phase on browsers. System testing proved that the chosen methodology was a good choice as there was no technical issues within the system and all the features work well together. Some small cosmetic issues were found.

5.4. Beta Testing

With the system testing complete, the decision was made to release a test server using Forge and Digital Ocean where few different people would test the application in their own time. Giving them a brief introduction to the project and listing them with the guide of what needs to be tested and looked at. This method is very effective as it provides cheap solution to test the end user.

Overall five people were involved in Beta testing where they provided their feedback and listed any potential issues within the system. The key elements to what they were looking at in the application were: login and registration, event creation, event editing and deletions, joining events, searching for events and editing account settings to change their profile.

Beta testing took a while and the results can be seen in the Evaluation section.

6. EVALUATION

Evaluation covers the project outcomes, beta evaluation as well as methodology and plan evaluation.

6.1. Beta Testing Evaluation

Beta Testing turned to be much more efficient than expected, overall the response of people was very positive. However, people were able to find few cosmetic issues in the system that were taken note of. One of the issues to mention was that if user submitted a description with no spaces, text would overlap into event creator information.

Some useful feature requests were also present from the users. One user mentioned that it would be possible to automatically recreate the event when it has finished and create it for next week or month, then the site would greatly benefit from it. Another feature mentioned was that it would be very effective when the user provides their location in the settings, and then searches for events it would provide a visual text message what events are near the user. One person raised an issue with how the user's status works in event page, they felt that when person selects that they are not able to make it to the event then this person should not be counted in the event at all. Because if the event is full but majority of people have their status set that they can't make it, it disables other people from joining. Event creator would need to check the event regularly and remove these members from the event. A potential new feature could be developed and enable users to join into waiting list when event is full.

Beta Testing provided useful feedback. But in order to release the application, a bit more testing needs to be done and more people need to access it before release. Additional some of the features and issues raised in Beta need to be taken care of before proper release.

6.2. Project outcomes

Looking at the project and the final result it can be said that the project went well. The application fits well with the problem that it was developed for, people are able to search and

create sport events in their local area. All of the core functionality was implemented and completed with success, very few requirements were not implemented. There was very positive reaction to the application during Beta Testing.

Use of different tools and technologies helped to developed the project, with that in mind a lot was also learned and discovered. Learning Laravel, using different libraries, APIs and implementing them in the system resulted in gaining some essential experience and skills for the future.

As for the Design and user experience, it feels that the system works well and looks well. User journey on the website proved simple, not complicated and the information on the pages are not confusing. Colour scheme fits well with the fonts and different icons used in the website.

Use of correct methodology and correct planning helped to make the project a success. Project could be improved by implementing all the features and taking a closer look at the feedback received from testing phase. Use of social media could have been used in order to gain some feedback on the design of the website.

Overall Project resulted in a well build, functional application that can be used in order to find or create sport events, users are able to join events, customize their settings, edit events or event provide their location for other members to see. Home page screen of the website can be seen in [Appendix 16].

6.3. Methodology Evaluation

Iterative and incremental development methodology proved to be the right choice for this project. It was possible to create all the features step by step and find any errors and bugs early while the features were implemented. Doing so saved time and provided solid structure while developing.

Using different methodology method in the project would result in spending more time while on testing and fixing all the errors in one stage. Iterative and incremental worked well as there

was good planning and design of the whole system and everything proved to work correctly while implementing feature.

6.4. Plan Evaluation

From start everything was plan and thought through, although some of features were developed and planned at the later stage. Still the overall plan of the project worked well. Decision of what to develop and use were justified and there was appropriate back up plans from beginning of the project.

The main plan was to develop application where people are able to search, join and create sport events and that was achieved. Project followed good time frame where core of the functionality was implemented.

Sometimes different features required more time than others to implement but that was expected. Plan was there to guide the project not to execute it with the perfection as it is very unrealistic approach. Design plan of the application worked particularly well as most of the pages were designed and look as they were planned to look. Some changes were made but it was more to do with some other factors like; not having enough data, statistics were planned to be implemented on the profile page for the users, but there was just not enough data to create them. Possibly in the future if other features are implemented this can be added on.

7. CONCLUSION

7.1. Report Summary

Report has followed a software development life cycle for developing a new application, using Iterative and incremental methodology to help guide the project. Report has included the different phases of the project starting from research, gathering requirements and analysis which can be seen in Concept Definition. Design section where the different plans regarding the user journey and user experience were made, different designs of the pages were investigated and evaluated. Also things like system design and data design was investigated. Development started soon after choosing the final designs, implementation section dives into how the application was created and what different technologies were used. Testing is another step of the software development life cycle and Sporty has came through testing phase with the head held high. Report has documented each step of the application and it helped to make some decisions by analyzing different risks and technologies.

7.2. Project Reflection

Project turned to be very enjoyable, time has gone so fast and lots of work and effort has been put into it. Overall feeling that the project was well planned, designed, implemented and tested is a positive reflection. There was a huge learning curve with both the back-end and front-end development, but it was definitely worth it. Learning how to use Laravel, Google Maps API has increased the skills set of my programming. Using the tools that are already developed, are well designed and maintain can have a great benefit to the project, I have learned that it is not always the best idea to develop everything from scratch, using current solutions that are well recognized in the industry will make the project much more efficient.

By developing Sporty I hope to make people interact more with each other through sport, and I feel that I have achieved that.

7.3. Role Reflection

Having all the control over the project, it meant that I had to fulfill different roles. From planning and being the Project Manager to being a designer and working on UI and UX.

Following with the development and the role of back-end and front-end developer to being QA Tester. I can honestly say that the most enjoyable role for me during the project was being a back-end developer. Working with handling the data and managing all the logic is something that I enjoy the most and this project has helped to increase my skills as a back-end developer. Front-end developer job was enjoyable too, but I feel that parts of design is my weakest point and the job required to do a bit of work with programming the design of the website.

7.4. Future

Future of Sporty is firstly to have a look at the results from the Beta Testing and improve the application based on the feedback received. Plan is to launch the application but it needs to go through a bigger testing phase first. There are a lot of features that are already on my mind if the application proves to have some interest. The point system where event created would be able to award users who came to the event with points, as well as enable users to rate the event after it has finished. Another feature that could be implemented is to allow people communicate across the site, so a notification / text chat feature. It would also be very beneficial if users would be able to create teams and leagues within their area and lastly integrating different social media features like login with different services.

There is just so much potential with the website, in the future Sporty is going to be released and the main focus for now would be to build some interest through social media and see where it gets to.

9. REFERENCES

1. Agile (2008) The agile movement. Available at: <http://agilemethodology.org/> (Accessed: 28 April 2016).
2. Archistico, E. (2009) Archistico font free by Archistico. Available at: <https://www.fontsquirrel.com/fonts/archistico> (Accessed: 28 April 2016).
3. Artisan, T. (2016) Artisan console - Laravel - the PHP framework for web artisans. Available at: <https://laravel.com/docs/5.2/artisan> (Accessed: 28 April 2016).
4. AWS (2016) Amazon web services (AWS) - cloud computing services. Available at: <http://aws.amazon.com/> (Accessed: 28 April 2016).
5. Bitbucket (2016) Bitbucket — the git solution for professional teams. Available at: <https://bitbucket.org/> (Accessed: 28 April 2016).
6. Blade, T. (2016) Blade Templates - Laravel - the PHP framework for web artisans. Available at: <https://laravel.com/docs/5.2/blade> (Accessed: 28 April 2016).
7. Canvas (2016) HTML5 canvas. Available at: http://www.w3schools.com/html/html5_canvas.asp (Accessed: 20 April 2016).
8. Carbon (2016) A simple PHP API extension for DateTime. Available at: <http://carbon.nesbot.com/docs/> (Accessed: 28 April 2016).
9. Catlin2015Hampton (2006) Sass: Sass basics. Available at: <http://sass-lang.com/guide> (Accessed: 28 April 2016).
10. Collections, T. (2016) Collections - Laravel - the PHP framework for web artisans. Available at: <https://laravel.com/docs/5.2/collections> (Accessed: 28 April 2016).
11. Composer (2016) Composer. Available at: <https://getcomposer.org/> (Accessed: 28 April 2016).
12. CSRF, N. (2015) Cross-site request forgery guide: Learn all about CSRF attacks and CSRF protection. Available at: <http://www.veracode.com/security/csrf> (Accessed: 28 April 2016).
13. DaFont (2016) Download fonts. Available at: <http://www.dafont.com/> (Accessed: 28 April 2016).
14. DBComparison (2014) SQLite vs MySQL vs PostgreSQL: A comparison of relational database management systems. Available at: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems> (Accessed: 28 April 2016).

15. DigitalOceanDigitalOcean™ (2016) Simple cloud computing, built for developers.
Available at: <https://www.digitalocean.com/> (Accessed: 28 April 2016).
16. Dunham, E. (2009) Free commercial fonts. Available at: <https://www.fontsquirrel.com/> (Accessed: 28 April 2016).
17. Eloquent (2016) Laravel Eloquent. Available at: <https://laravel.com/docs/5.2/eloquent> (Accessed: 28 April 2016).
18. Engebretson, A. (2015) Laravel collective. Available at:
<https://laravelcollective.com/docs/5.2/html> (Accessed: 28 April 2016).
19. Forge (2016) Laravel forge - instant PHP servers. Available at:
<https://forge.laravel.com/> (Accessed: 28 April 2016).
20. Glyphicons (2016) Bootstrap. Available at: <http://getbootstrap.com/components/> (Accessed: 28 April 2016).
21. Google Fonts (2016) Google fonts. Available at: <https://www.google.com/fonts> (Accessed: 28 April 2016).
22. Gulp (2016) Easy to use. Available at: <http://gulpjs.com/> (Accessed: 28 April 2016).
23. Hern, A. (2015) Smartphone now most popular way to browse internet – Ofcom report.
Available at: <https://www.theguardian.com/technology/2015/aug/06/smartphones-most-popular-way-to-browse-internet-ofcom> (Accessed: 28 April 2016).
24. Homestead, T. (2016) Laravel homestead - Laravel - the PHP framework for web artisans. Available at: <https://laravel.com/docs/5.2/homestead> (Accessed: 28 April 2016).
25. Intervention (2016) Intervention image - introduction. Available at:
<http://image.intervention.io/> (Accessed: 28 April 2016).
26. Laracasts, all of (2016) The best Laravel and PHP Screencasts. Available at:
<https://laracasts.com/> (Accessed: 28 April 2016).
27. LLC, G.R. (2010) Free graphic resources for everyone. Available at:
<http://www.freepik.com/> (Accessed: 28 April 2016).
28. LP, is. (2016) Stock photos, royalty-free images, video & music clips - iStock.
Available at: <http://www.istockphoto.com/> (Accessed: 28 April 2016).
29. Mandrill, A. rights (2016) Transactional Email from MailChimp. Available at:
<https://www.mandrill.com/> (Accessed: 28 April 2016).
30. MERA (2016) Iterative and incremental development. Available at:
<https://www.mera.com/services/processes/iterative> (Accessed: 28 April 2016).

31. Migrations, T. (2016) Database: Migrations - Laravel - the PHP framework for web artisans. Available at: <https://laravel.com/docs/5.2/migrations> (Accessed: 28 April 2016).
32. Nguyen, T. (2016) User persona creator by Xtensio (it's free!). Available at: <https://xtensio.com/user-persona/> (Accessed: 29 April 2016).
33. Node (2016) Node.Js. Available at: <https://nodejs.org/en/> (Accessed: 28 April 2016).
34. Otwell, T. (no date) The PHP framework for web artisans. Available at: <https://laravel.com/> (Accessed: 28 April 2016).
35. Participation Report (2015) Available at: <http://www.physicalactivitycouncil.com/pdfs/current.pdf> (Accessed: 28 April 2016).
36. Parzuchowski, M. (2016) Boy child kid free stock photo. Available at: <https://stocksnap.io/photo/RSP6NAGL6B> (Accessed: 28 April 2016).
37. Pexels (2016) Free high quality photos · Pexels. Available at: <https://www.pexels.com/> (Accessed: 28 April 2016).
38. Pirttijärvi, D.M.P. (2015) Javascript identicon generator. Available at: <https://jidenticon.com/> (Accessed: 28 April 2016).
39. prototyping (2016) SDLC - software prototype model. Available at: http://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm (Accessed: 28 April 2016).
40. Pty2016SitePoint (2015) The best PHP framework for 2015: SitePoint survey results. Available at: <http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/> (Accessed: 28 April 2016).
41. Rackspace (2016) Transactional Email API service for developers by Rackspace. Available at: <https://www.mailgun.com/> (Accessed: 28 April 2016).
42. Rackspace (2016) UK: Homepage spring 2015. Available at: <http://www.rackspace.co.uk/> (Accessed: 28 April 2016).
43. RAD (2016) SDLC - RAD model. Available at: http://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm (Accessed: 28 April 2016).
44. Roboto (2016) Google Fonts. Available at: <https://www.google.com/fonts/specimen/Roboto> (Accessed: 28 April 2016).
45. Stacksnap (2016) Beautiful free stock photos (CC0). Available at: <https://stocksnap.io/> (Accessed: 28 April 2016).
46. SweetAlert (2000) SweetAlert. Available at: <http://t4t5.github.io/sweetalert/> (Accessed: 28 April 2016).

47. System testing (2016) ISTQB exam certification. Available at:
<http://istqbexamcertification.com/what-is-system-testing/> (Accessed: 28 April 2016).
48. TeamSnap (2005) TeamSnap for teams, leagues, clubs and associations. Available at:
<https://www.teamsnap.com/> (Accessed: 28 April 2016).
49. Testing, J. (2013) Types of system testing - Enfocus solutions Inc. Available at:
<http://enfocussolutions.com/types-of-system-testing/> (Accessed: 28 April 2016).
50. The free sports team management App (2008) Available at: <https://teamer.net/>
(Accessed: 28 April 2016).
51. Tools and Performance (2015) Simulate mobile devices with device mode | web tools - Google developers. Available at: <https://developers.google.com/web/tools/chrome-devtools/iterate/device-mode/?hl=en> (Accessed: 28 April 2016).
52. Vagrant (2016) Vagrant by HashiCorp. Available at: <https://www.vagrantup.com/>
(Accessed: 28 April 2016).
53. Validation Rules, T. (2016) Validation - Laravel - the PHP framework for web artisans.
Available at: <https://laravel.com/docs/5.2/validation#available-validation-rules>
(Accessed: 28 April 2016).
54. VM (2016) Oracle VM VirtualBox. Available at: <https://www.virtualbox.org/> (Accessed: 28 April 2016).
55. Waterfall development Methodology (2006) Available at:
http://learnaccessvba.com/application_development/waterfall_method.htm
(Accessed: 28 April 2016).
56. Who plays sport? (2015) Available at: <http://www.sportengland.org/research/who-plays-sport/> (Accessed: 28 April 2016).

10. APPENDICES

Appendix 1

Requirement #: 1	
Priority	1
Description	Allow users to register and create account in web application
Rationale	Enable users to use all features within system when logged in
Dependencies	User needs to provide information in order to create their account
Fit Criterion	Enable users to login to the system, display confirmation message on successful account creation.

Requirement #: 2	
Priority	1
Description	Allow users to login to the system
Rationale	To enable full use of web application and users account page
Dependencies	User needs to create account before logging in
Fit Criterion	Access account with username and password. Display log-in details (users name or username) on the current viewing page.

Requirement #: 3	
Priority	1
Description	Allow users to reset / retrieve password for their account
Rationale	To enable full use of web application and users account page
Dependencies	User needs to create account before resetting password
Fit Criterion	Provide email address associated with users account, email is sent with instructions on how to reset password.

Requirement #: 4	
Priority	1
Description	Allow users to delete account from the system
Rationale	To disable use of all the features and users account
Dependencies	User needs to create account before deleting it
Fit Criterion	Delete all data and information associated with users account, before deleting display alert warning message to confirm users action.

Requirement #: 5	
Priority	1
Description	Allow users to logout from the system
Rationale	This will end current session and disable use of all features
Dependencies	User needs to login before logging out
Fit Criterion	Display successful logout message when session has ended

Requirement #: 6	
Priority	1
Description	Allow users to create sport events
Rationale	To search for people who want to participate in it
Dependencies	User needs to login before creating event
Fit Criterion	User provides details about the event when creating it. On successful creation event is displayed on search page.

Requirement #: 7	
Priority	1
Description	Create a search bar and search options

Rationale	To allow users to search for events
Dependencies	No dependencies
Fit Criterion	Return a list of events based on search selection, suggest events if no events found

Requirement #: 8	
Priority	1
Description	Allow users to sign up for events
Rationale	
Dependencies	User needs to be logged in to the system
Fit Criterion	User is listed as a signed up user for the event, creator of the event receives information that some one has signed up

Requirement #: 9	
Priority	1
Description	Allow users to create a profile page
Rationale	Other people can view your page and contact you
Dependencies	User needs to be logged in to the system to access account settings
Fit Criterion	User provides details about themselves like: skill level, age, location, sports they are interested in and nickname, email etc

Requirement #: 10	
Priority	2
Description	Allow users to search for other users
Rationale	So that they can add each other as friends
Dependencies	User needs to be logged in to the system to search for other users

Fit Criterion	Search other users by typing their username in the url, this will direct to users profile page where current user can send friend request
----------------------	---

Requirement #: 11	
Priority	1
Description	System will store event information about the users
Rationale	To show users record of which events they participated in
Dependencies	No Dependencies, access users profile page to see event history
Fit Criterion	List of events that the user participated in is displayed on profile page, sorted by date, only 3 visible, rest is shown when pressed 'Show All'

Requirement #: 12	
Priority	1
Description	Make web application responsive
Rationale	So that users can access system on different mobile devices
Dependencies	User Interface, experience design
Fit Criterion	Use of framework to fulfil media queries to specify different devices and display correct style for the device.

Requirement #: 13	
Priority	1
Description	System will allow event creator to cancel the event
Rationale	So that users will be aware if event no longer takes place
Dependencies	User needs to be logged in and needs to be event creator
Fit Criterion	When event is cancelled, all sign up users are notified and the data for the event is no longer displayed but it is stored in Database.

Requirement #: 14

Priority	1
Description	System will allow event creator to change details about the event
Rationale	So that users will be aware if event information are changed
Dependencies	User needs to be logged in and needs to be event creator
Fit Criterion	Notification is send to signed up users about new information for the event.

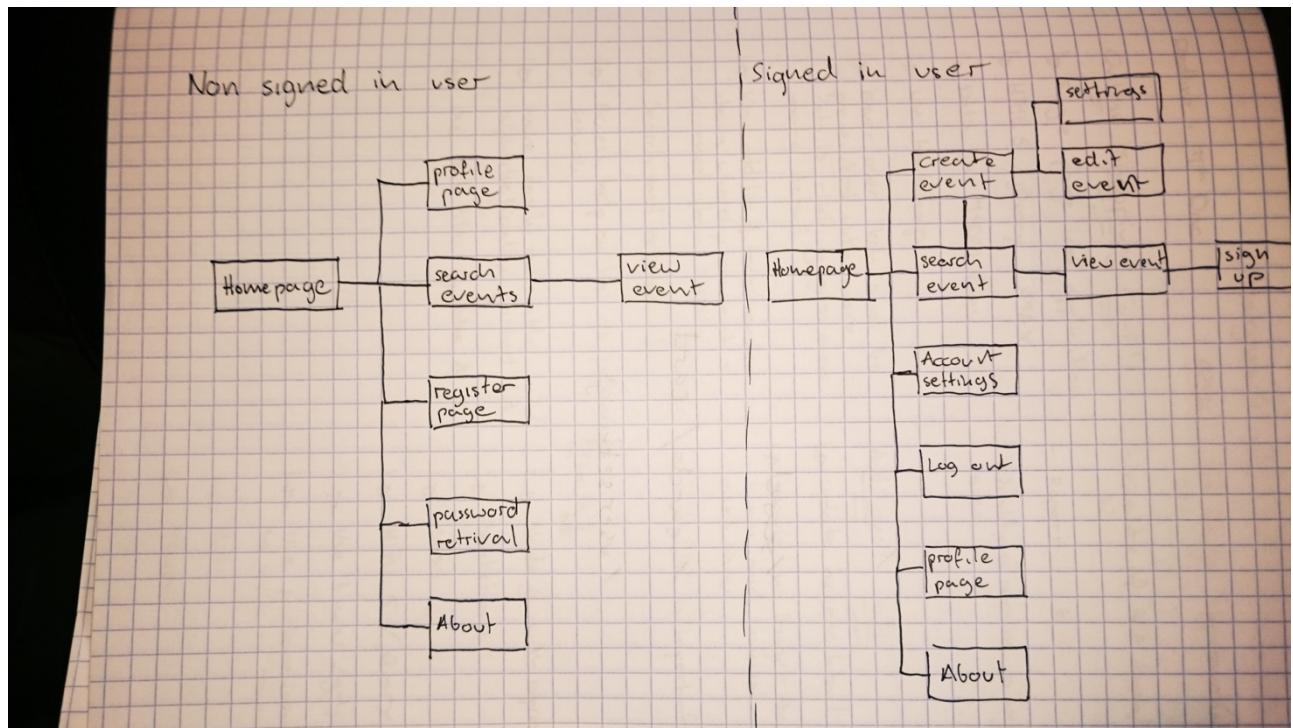
Requirement #: 15

Priority	1
Description	System will allow event creator to contact all signed up users
Rationale	To send important information or news about event
Dependencies	User needs to be logged in and needs to be event creator
Fit Criterion	Event creator will write a message and send it to everyone who is participating in the event, display successful message when done

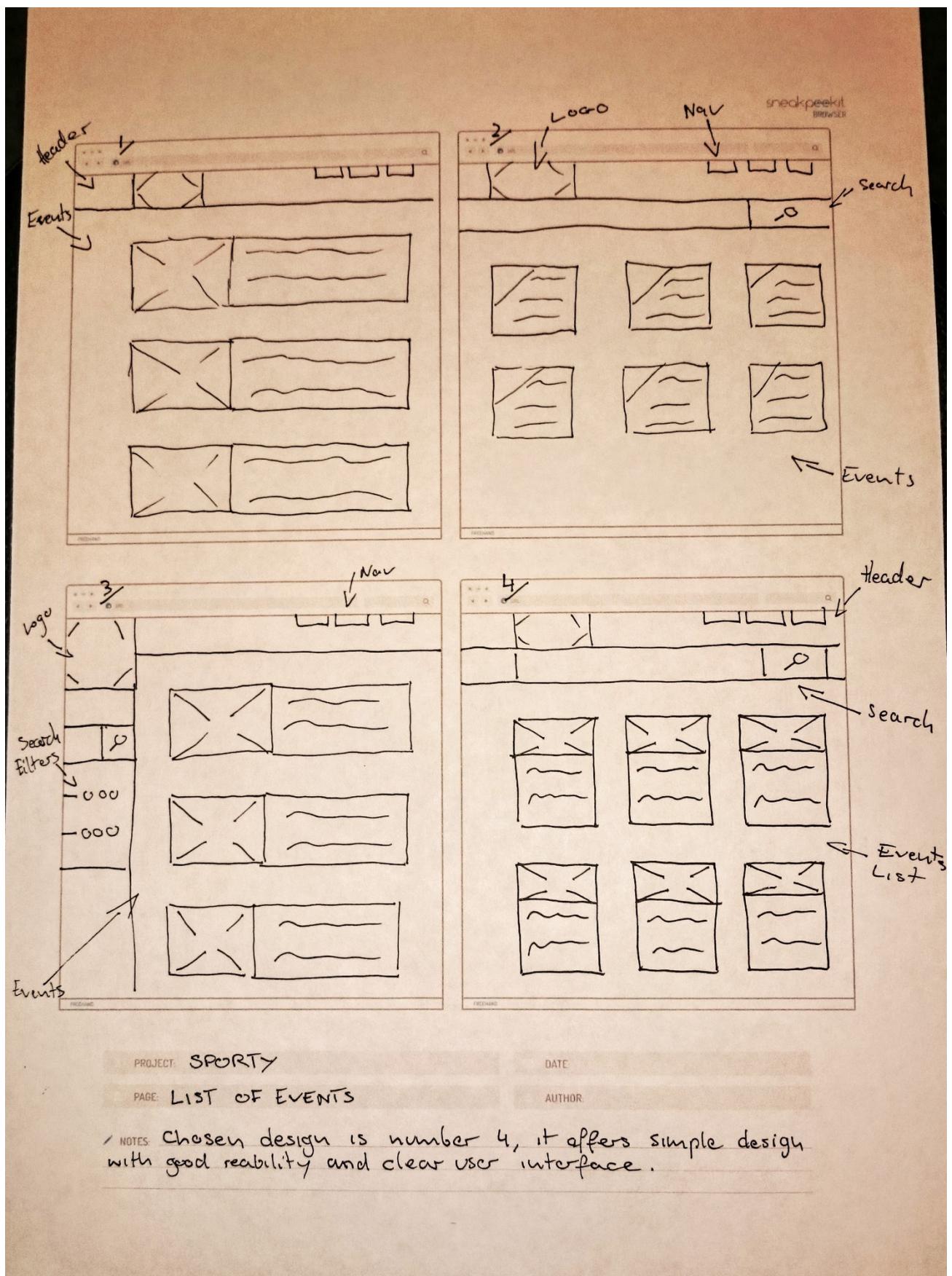
Requirement #: 16

Priority	1
Description	System will allow signed up user to drop/resign from event
Rationale	So that event creator is aware that the user is no longer available
Dependencies	User needs to be logged in
Fit Criterion	Confirmation message is displayed to the user that he/she has dropped from the event

Appendix 2



Appendix 3

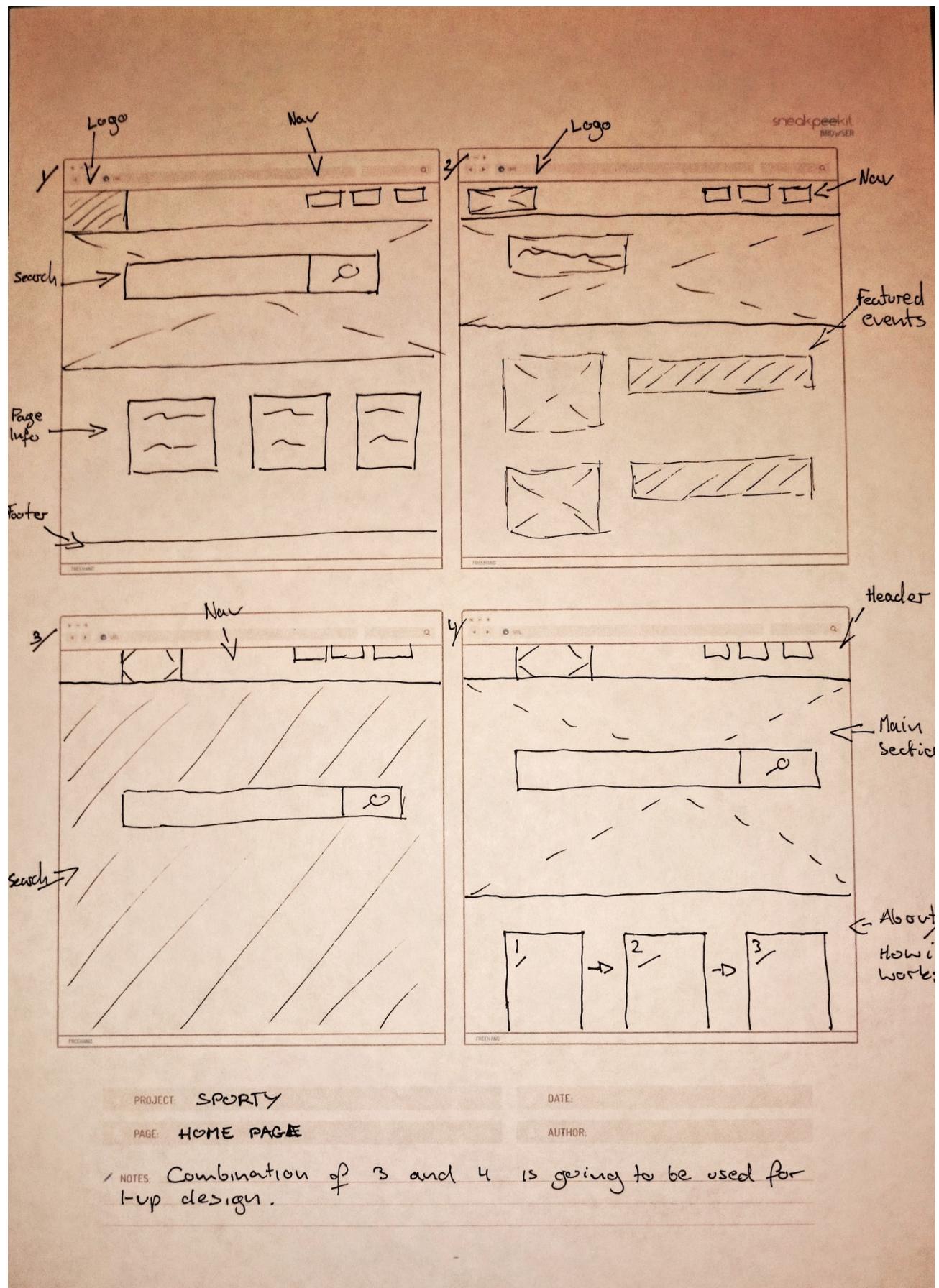


Four hand-drawn wireframe designs for an event page:

- Design 1:** Features a header, a large central image, a sidebar with 'Event info' (containing a 'JOIN' button), and a sidebar with 'Help'.
- Design 2:** Features a header, a large central image, a sidebar with 'Main info' (containing a 'JOIN' button), and a sidebar with 'Additional info'.
- Design 3:** Features a header, a large central image, a sidebar with 'Main info' (containing a 'JOIN' button), and a sidebar with 'Additional info'.
- Design 4:** Features a header, a large central image, a sidebar with 'Event info' (containing a 'JOIN' button), and a sidebar with 'Author info'.

Below the wireframes:

- PROJECT: SPORTY
- PAGE: EVENT PAGE
- NOTES: Chosen design is number 4
- DATE: _____
- AUTHOR: _____



Header

Profile info

Users history
of events

event thumbnails

sneakpeekit
BROWSER

Profile info

stats

History

Header

Profile info

history

event thumbnails

Header

Profile info

stats

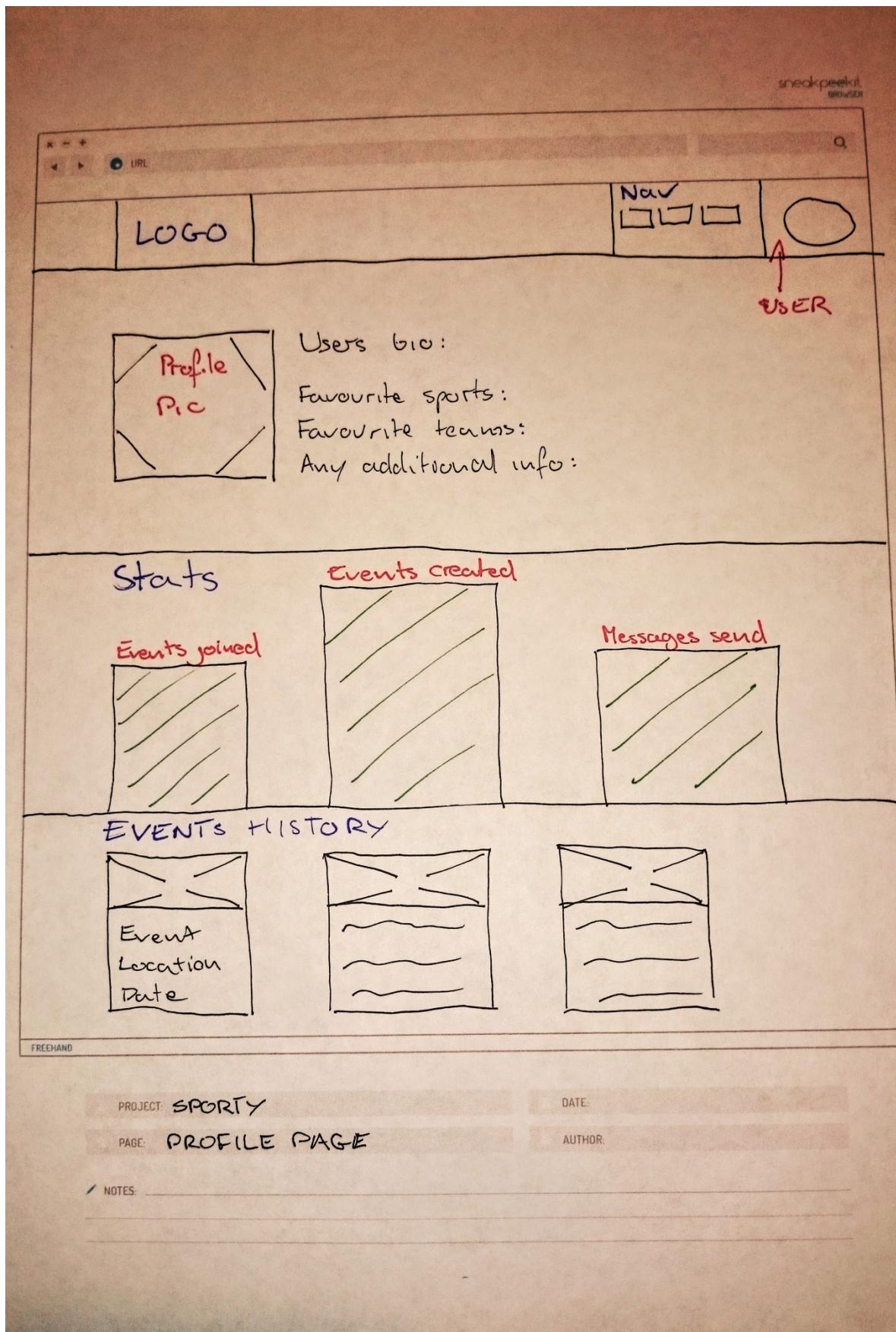
event thumbnails

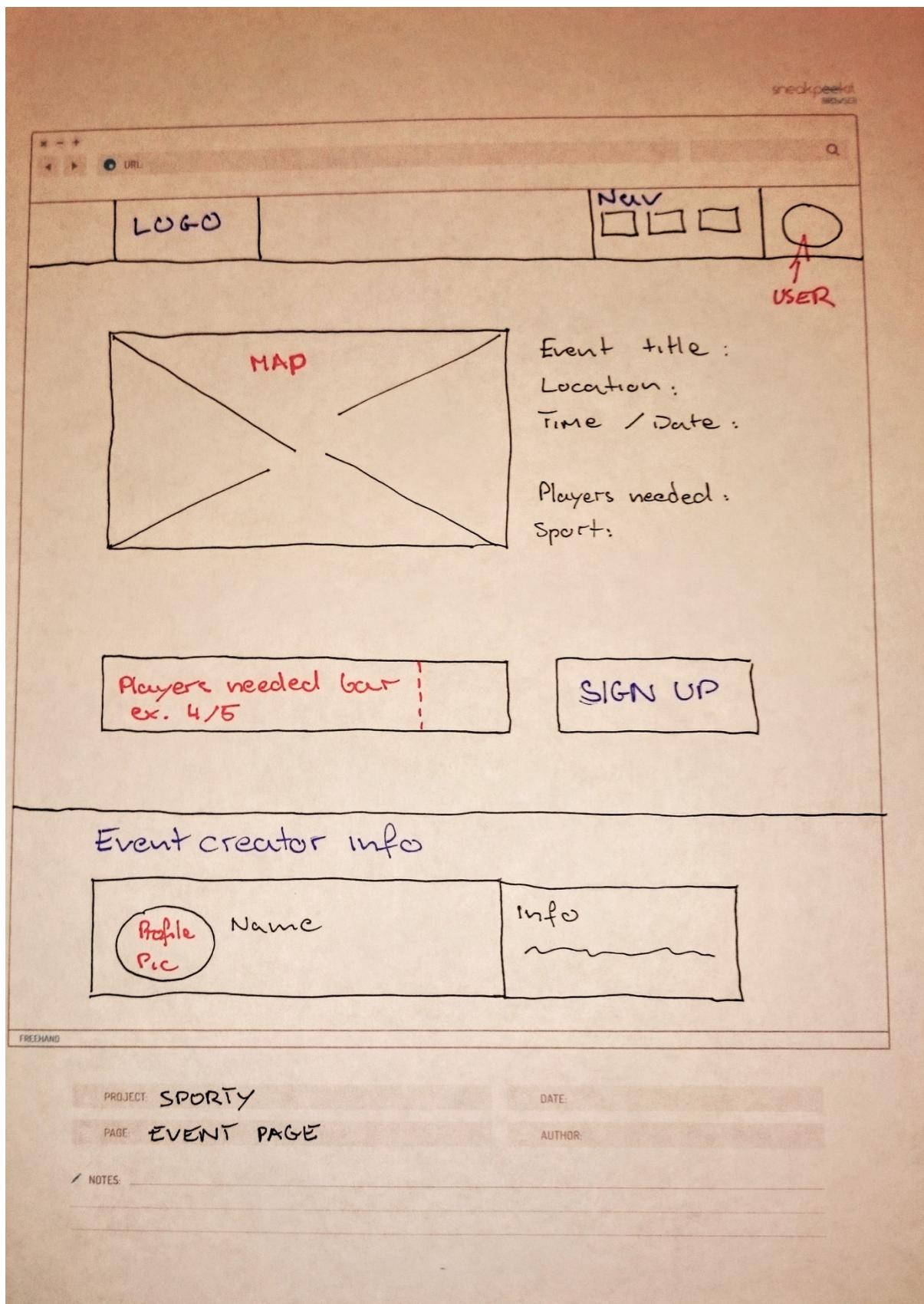
PROJECT: SPORTY

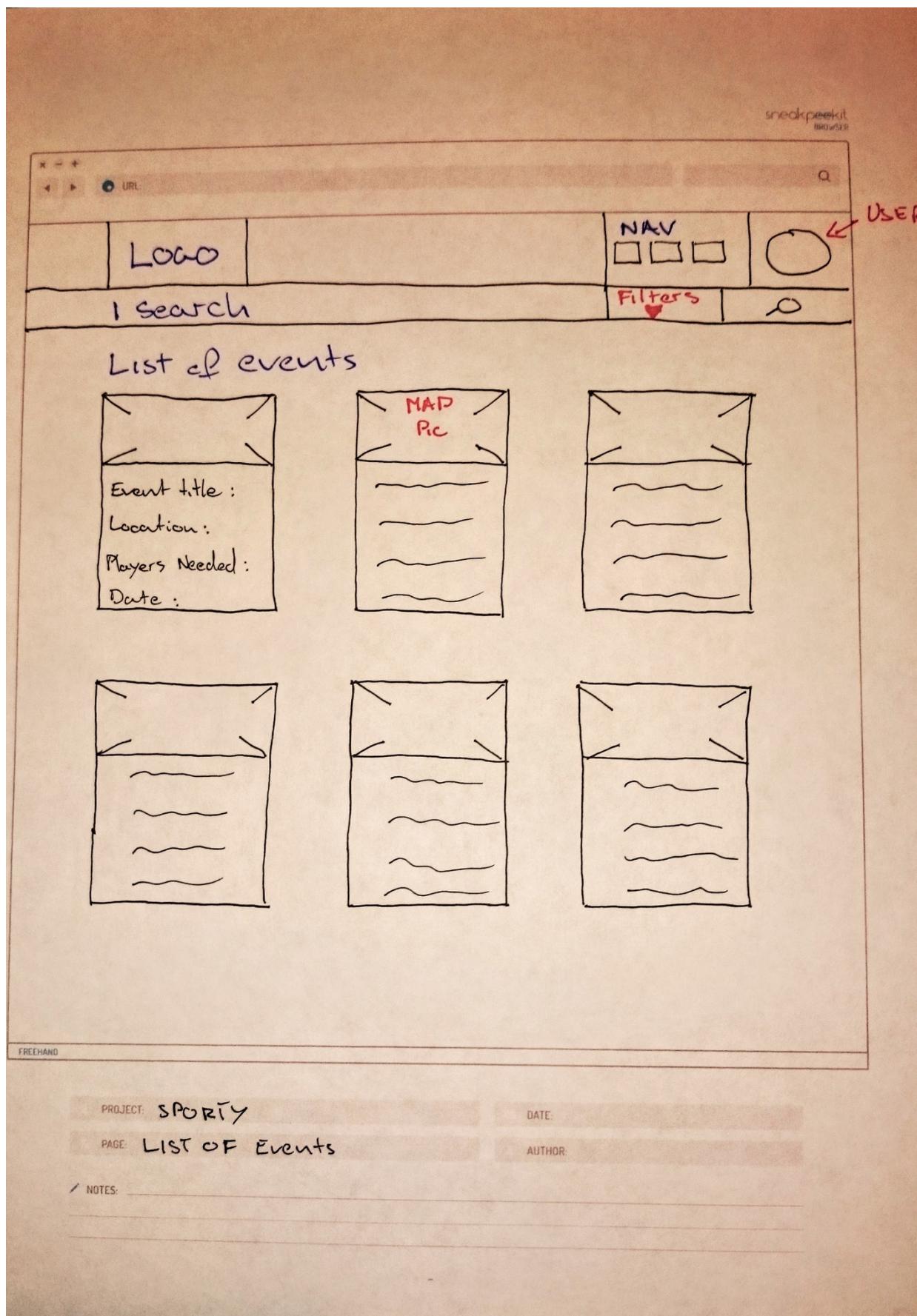
PAGE: PROFILE PAGE

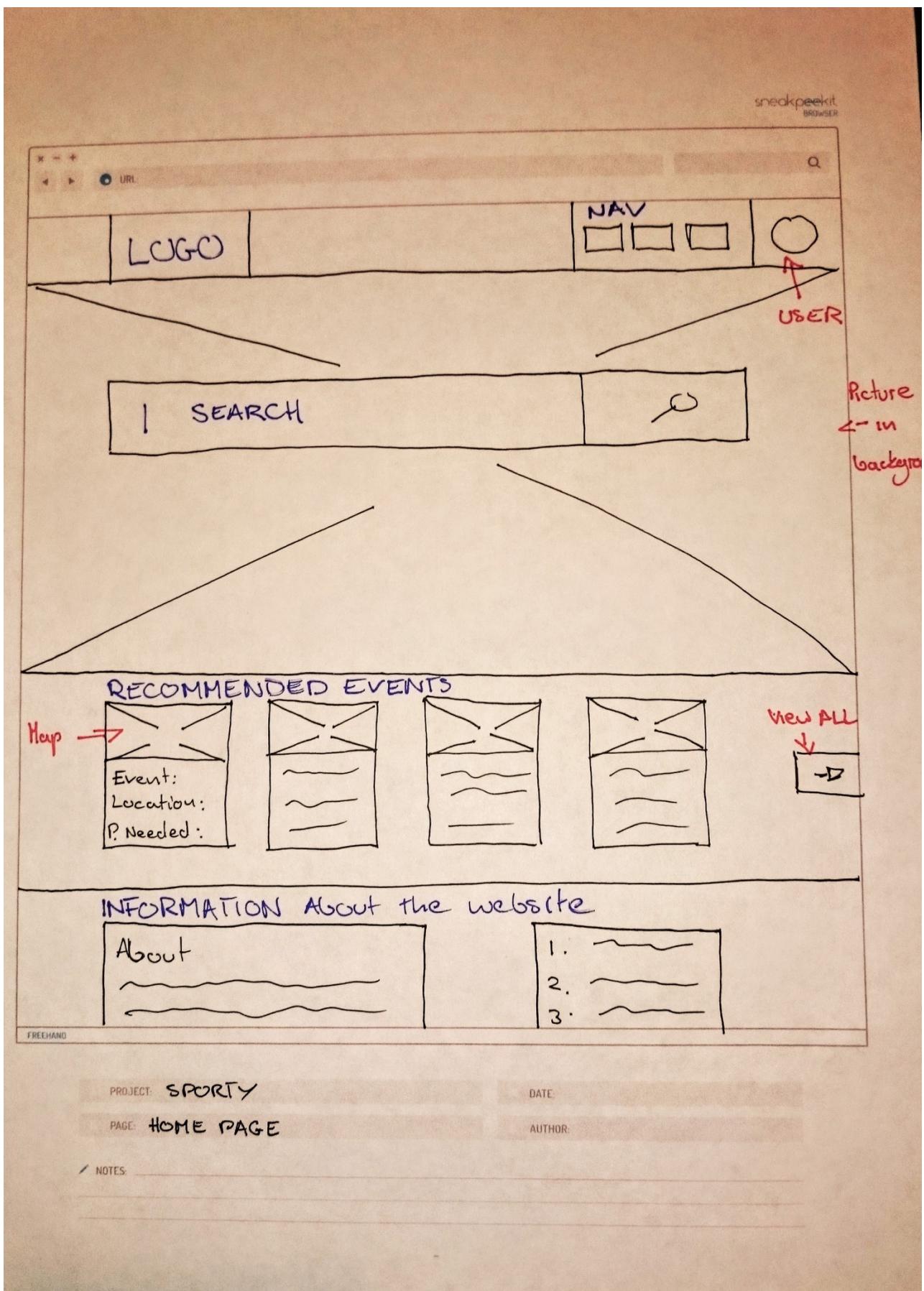
NOTES: Chosen design is number 2

Appendix 4

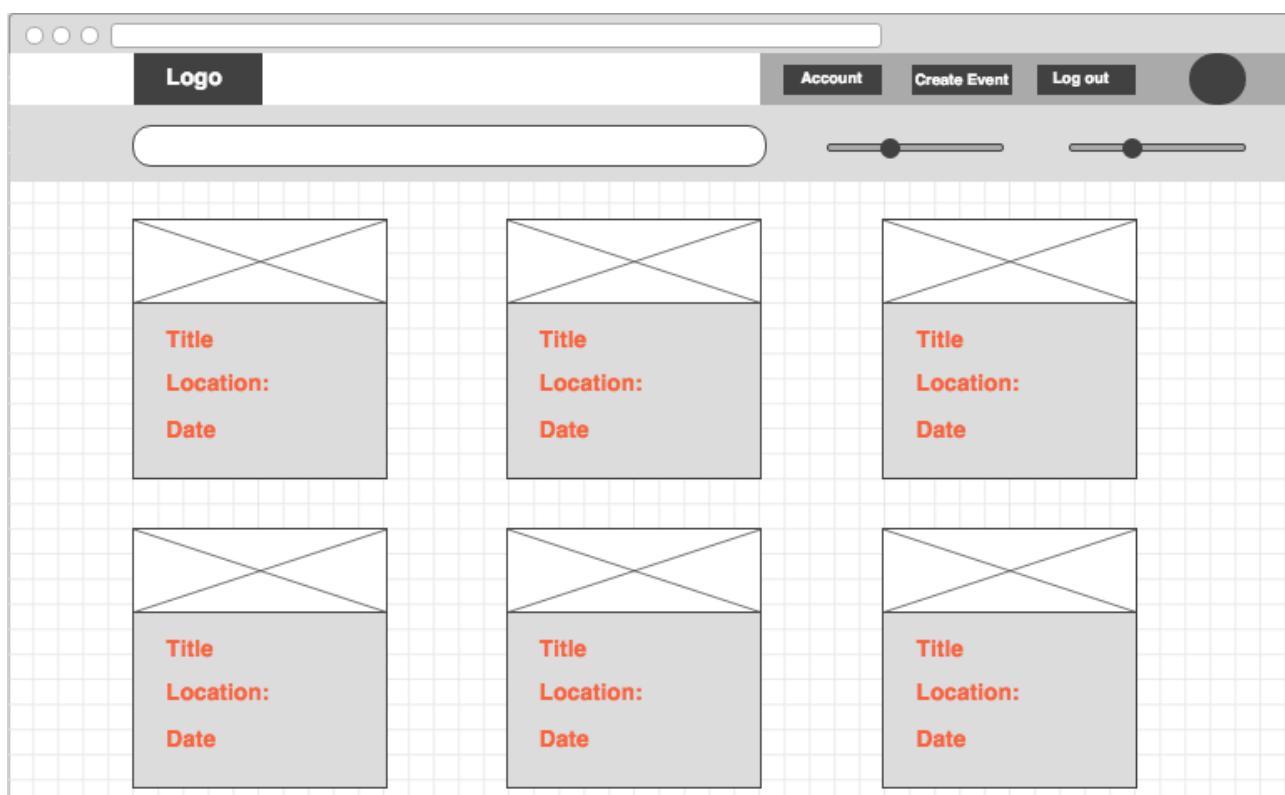
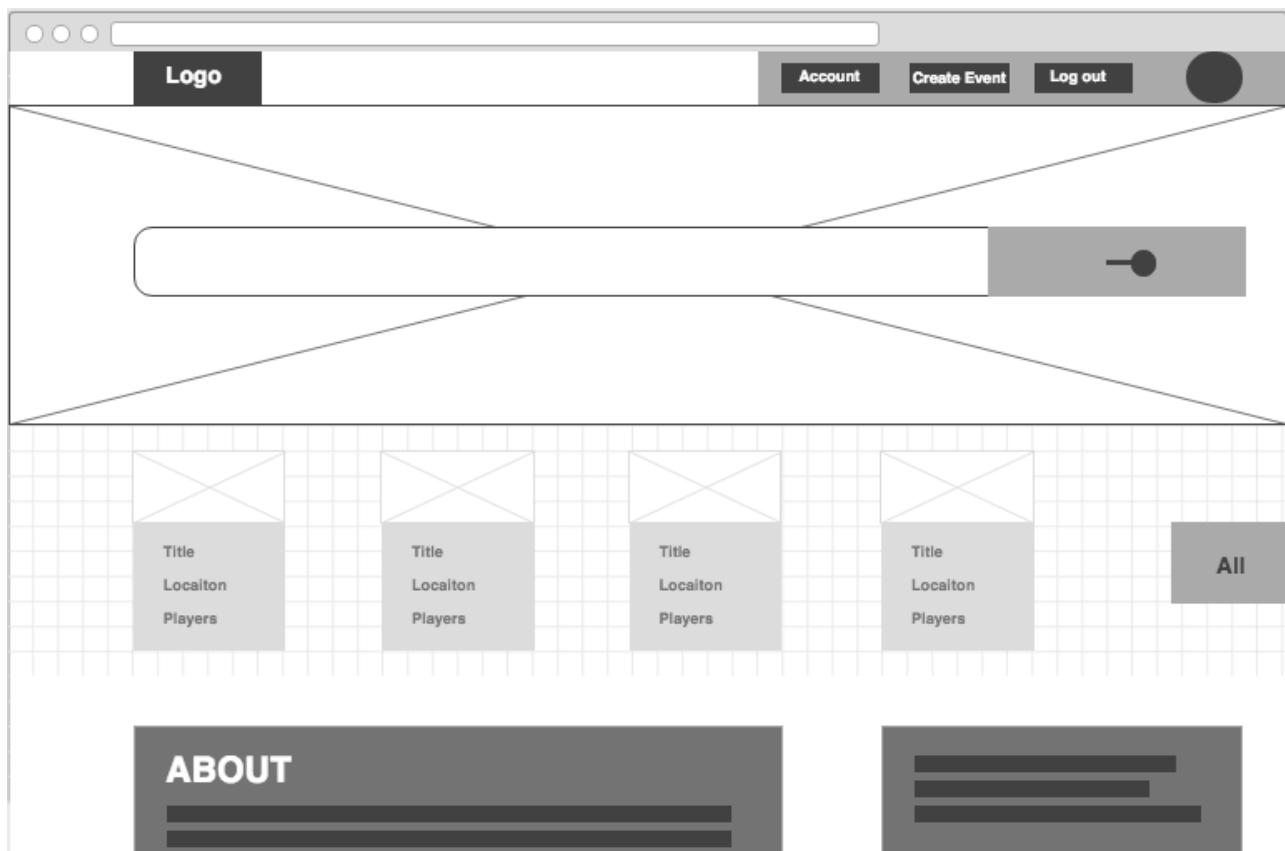


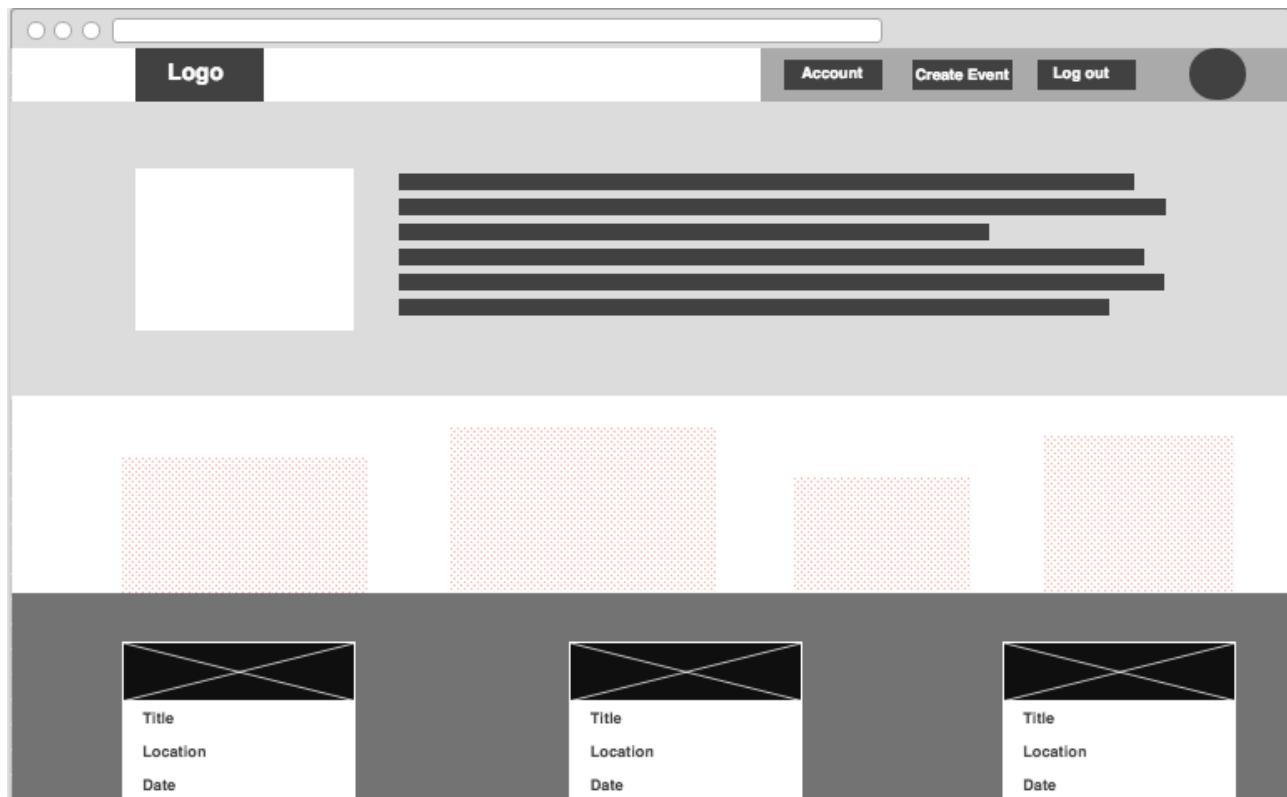
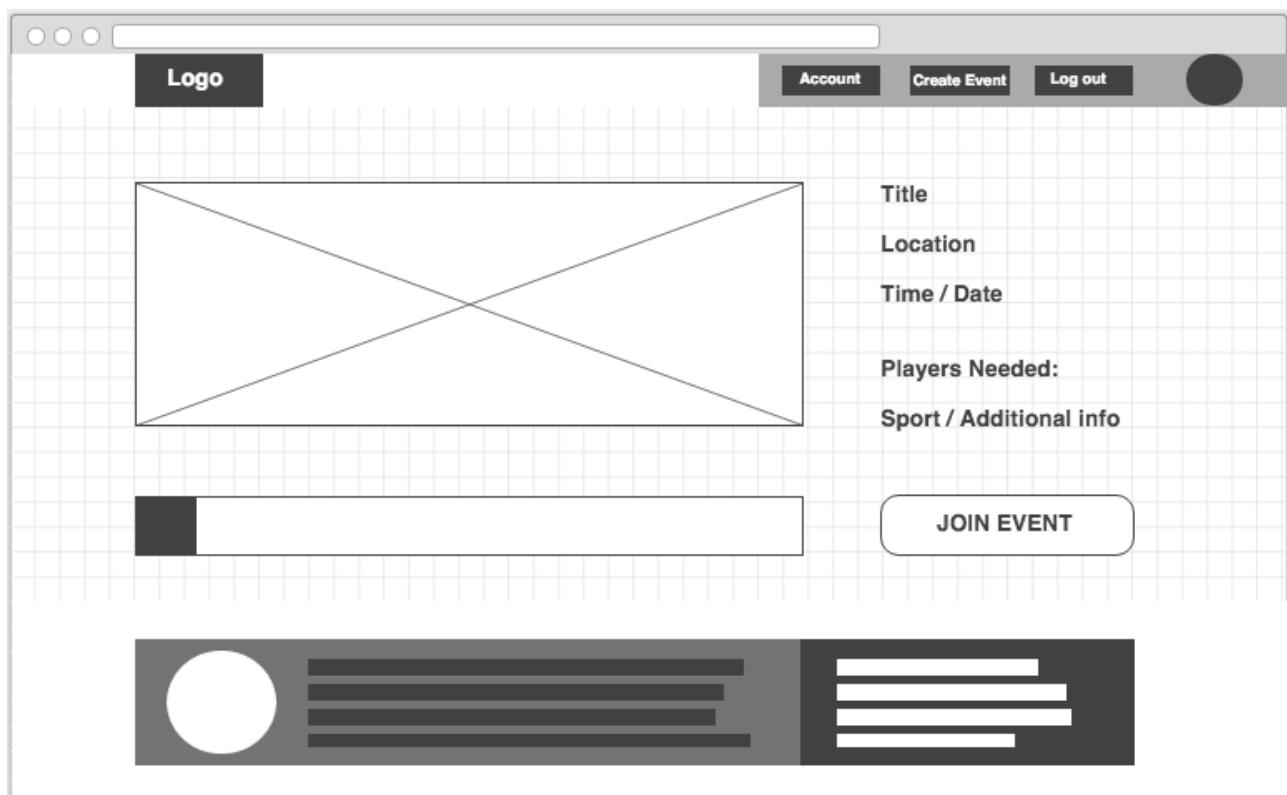




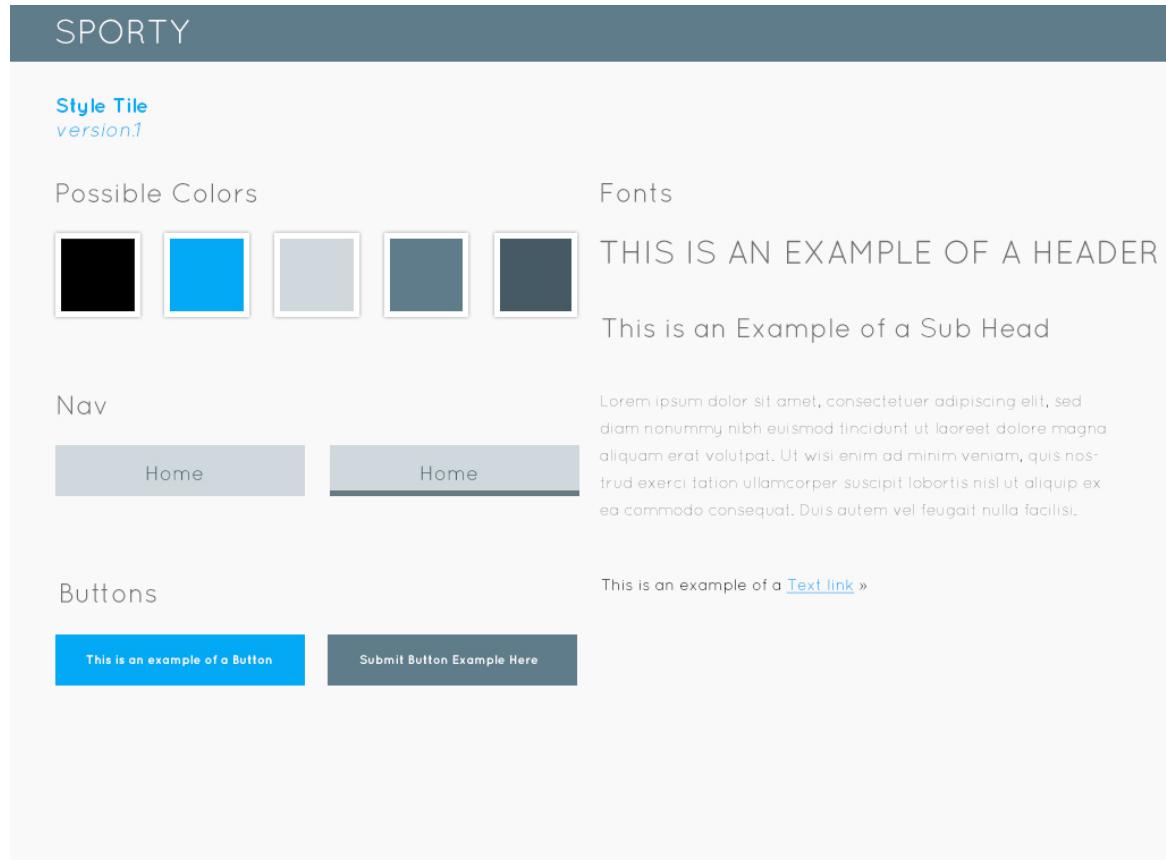
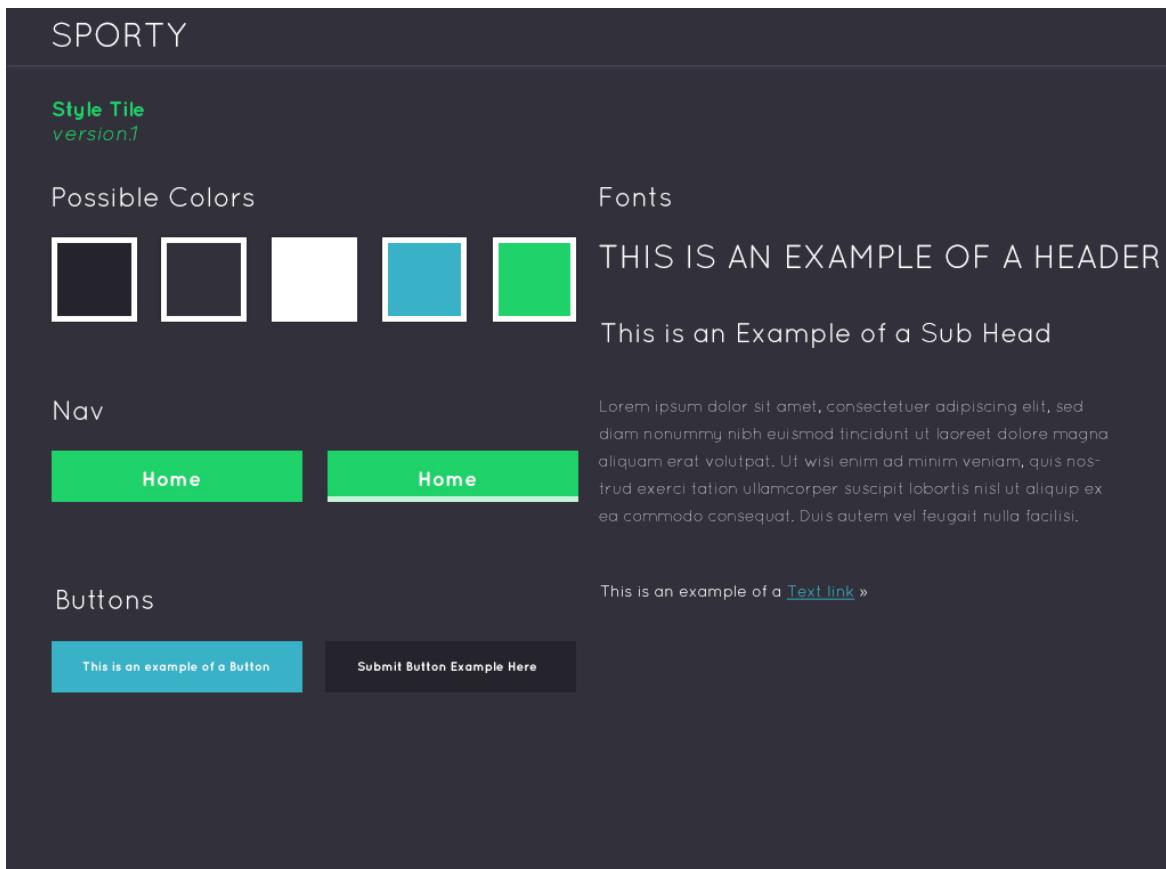


Appendix 5

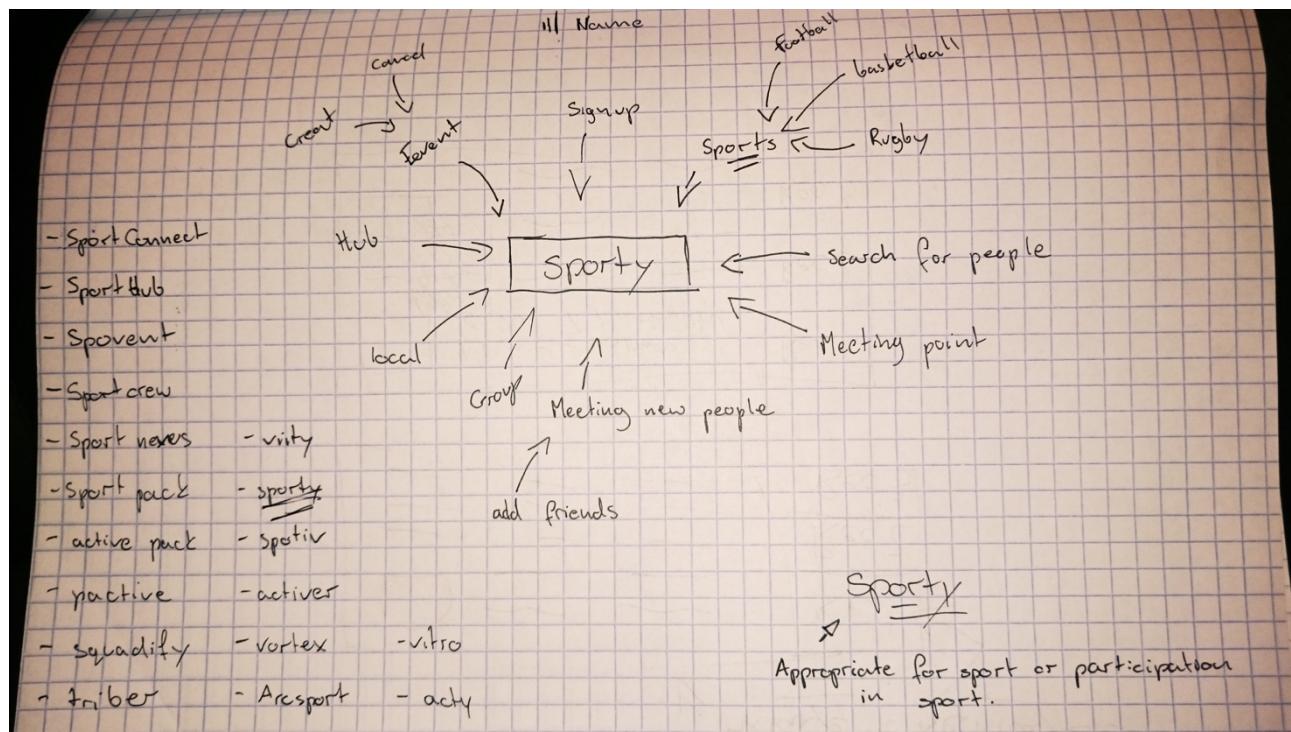




Appendix 6



Appendix 7



Appendix 8

Adam Rogers



Personal Trainer Athlete Gamer

Bio

Adam was born in Lisburn where he lived with his parents, he attended different colleges where he completed his courses and gained knowledge about his jobs. He moved to Belfast where he works in local Gym as Personal Trainer. Adam is passionate about keeping his body in shape and eating properly.

Adam also enjoys playing console games in his free time, he has participated in couple of FIFA tournament but he never got far in the competition.

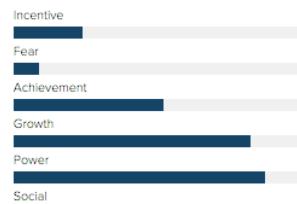
"It's not whether you get knocked down; it's whether you get up."

Age: 28
Work: Personal Trainer
Family: Married
Location: Belfast

Goals

- Meet new people
- Participate in Rampage obstacle course run
- Loose body fat
- Start organising sport events

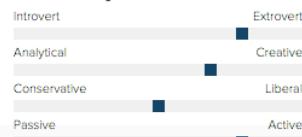
Motivations



Preferred Channels



Personality



Tom McMullan



Developer Gamer Musician

Bio

Tom was born in Belfast where he lived with his parents, he graduated from University of Ulster. Tom has a lot of experience in his field as he worked for many companies, at the moment Tom is working for local web design studio where he spending his days developing and solving problems for different clients and projects.

Tom is a member of a band where he plays guitar, his band plays rock music. Tom also likes to play console games with his friends or online, his favourite game is Metal Gear Solid series.

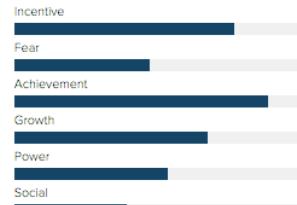
"Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live."

Age: 32
Work: Back-end web developer
Family: Married - 2 kids
Location: Belfast

Goals

- Be more active
- Start participating in sport activities
- Learn to Ski
- Meet new people

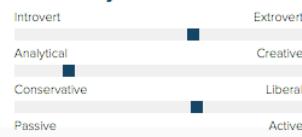
Motivations



Preferred Channels



Personality



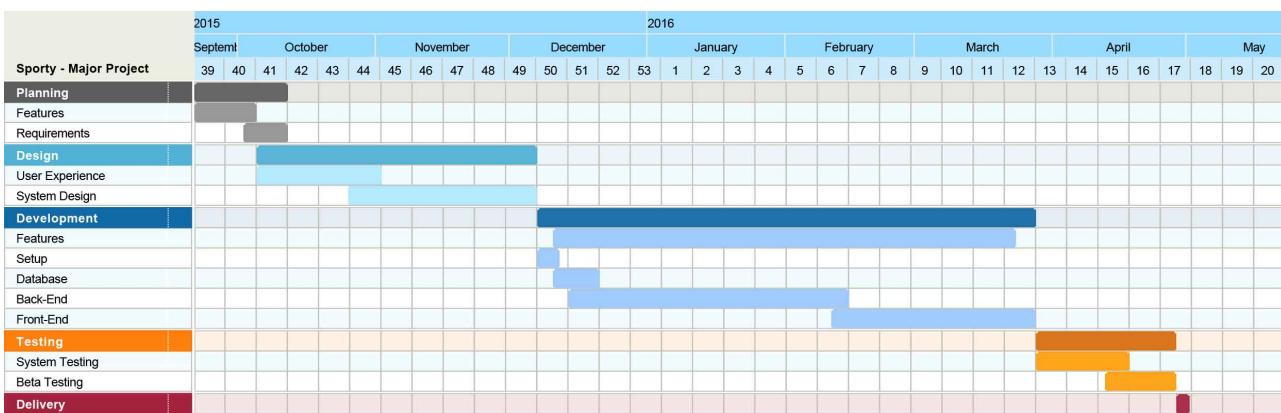
Appendix 9

Design		
Requirement	Priority	Description
1	1	Create a clean a simple design to navigate the website easily
2	3	Website needs to have a good colour scheme to make it look professional and attractive
3	1	Website needs to be usable and accessible from different devices
4	1	Website needs to work on different major browsers (Chrome, Firefox, Safari, IE11+)
5	2	Website should use icons to minimise text on screen and make the website more appealing

Usability		
Requirement	Priority	Description
1	1	Website needs to protect user's data
2	3	Website needs to take regular backups of data and be secured
3	2	Website needs to have good accessibility to remove the barriers of interaction or access to the website by people with disabilities

Performance		
Requirement	Priority	Description
1	1	Website needs to run smoothly with no errors or crashes
2	1	Website needs to be accessible online at all times
3	1	Website should be fast and load within 1.5sec
4	1	Website should handle at least 100 users at the same time, additional servers should be used if the numbers rise and exceed the expectation
5	2	Website should be easily installed, maintained and be transferable as well as extendable. Use of environmental variables can be useful to distinguish between local development and production.

Appendix 10



Appendix 12



Appendix 13



Appendix 14

LICENSE TERMS AND CONDITIONS

flaticon

FREE LICENSE (WITH ATTRIBUTION)

This license allows you to use for free any of Freepik contents for your projects as long as they are attributed to their author in the definitive project "designed by Freepik".

How to attribute contents?

- For web usage: By placing a link with the text "designed by Freepik" in a visible spot, so Freepik's authorship is noticeable.
- Uses different to web: If possible, the text "designed by Freepik" must be written next to Freepik Contents, if it's not possible, the attribution must be placed in the credits or acknowledgements section.

Where you can use Freepik contents:

- Website.
- Software, applications, mobile.
- Printed and digital media (magazines, newspapers, books, cards, labels, CD, DVD, films, television, video, e-mail).
- Advertisement and promotional items.
- Presentation of products and public events.
- Multimedia.
- Ornamentation (either private or public).

What you CAN DO:

- You have the non-exclusive, non-transferable, non-sublicensable right to use the licensed material an unlimited number of times in any and all media for the commercial or personal purposes listed above.
- You may alter and create derivative works.
- Your rights to the Licensed Material are worldwide.

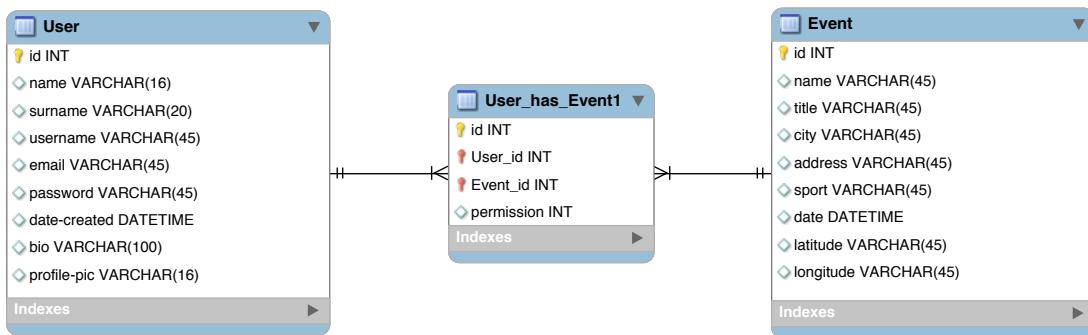
What you CAN NOT DO:

- Sublicense, sell or rent any contents (or a modified version of them).
- To distributed contents unless expressly authorized.
- To include the contents in an online or offline database or file.
- To offer Freepik designs (or their modified versions) for download.

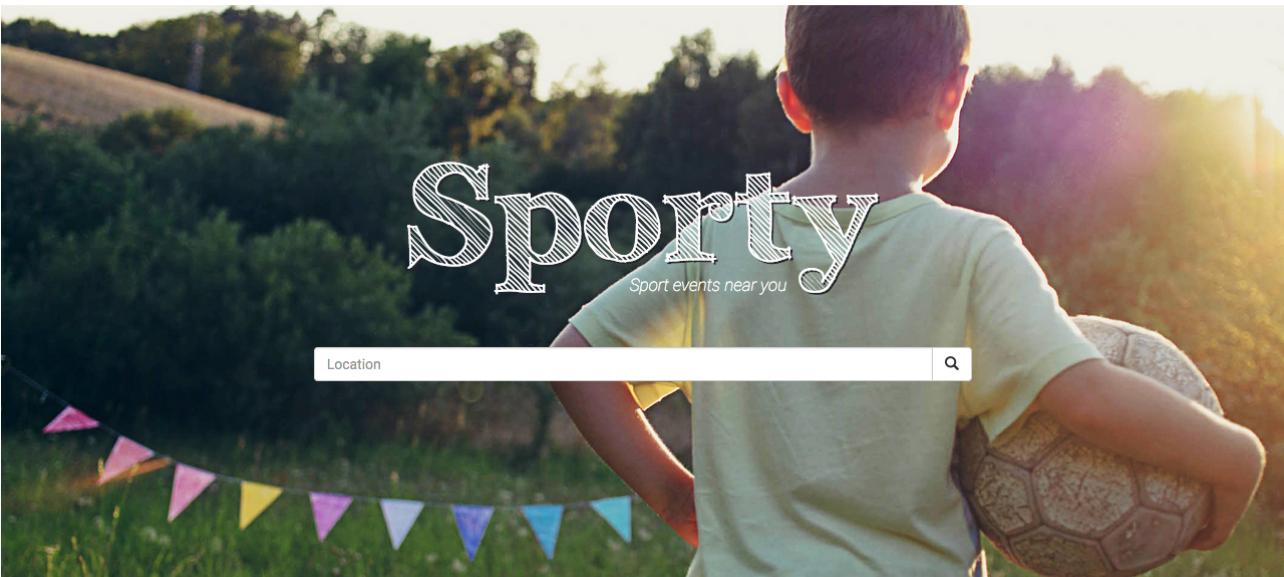
Graphic Resources LLC
 Commercial Registry of Málaga, volume 4994, sheet 217, page number MA-113059, with Tax Number B-93183366 and registered office at 16 Fernán Núñez Street, Office number 4, 29002 Málaga, Spain.



Appendix 15 – ER Diagram version 1



Appendix 16



The screenshot shows the homepage of the Sporty web application. At the top right are links for "Events", "Login", and "Register". A hexagonal logo with a stylized letter "S" is located at the top left. The main header features the word "Sporty" in large, bold, white, hand-drawn style letters, with the subtitle "Sport events near you" in smaller text below it. Below the header is a search bar with the placeholder "Location" and a magnifying glass icon. A horizontal row of various sport-related icons follows. The main content area has a dark background and features a large, bold heading "How it works" in white. Below this are three white rectangular boxes, each containing an icon and a title: "Create Events" (plus sign), "Search and Join" (magnifying glass), and "Meet and Play" (location pin). Each box also contains a brief description of the feature. At the bottom of the page, there's a "About Us" section with a paragraph of text, followed by copyright information and footer links.

Events Login Register

Sporty
Sport events near you

Location

How it works

Create Events

Creating events couldn't be simpler! Login to create sport event, provide necessary information and let other people find it.

Search and Join

Search sport events by location, filter through results and join the ones that suit you. Check event info and update your status if needed!

Meet and Play

Meet with other people who have joined at the events location, it is time to play and enjoy the sport! View your events in your profile page.

About Us

Sporty is a web application that enables users to search for people in order to participate in different sport events together. Users are able to create their own events and enjoy the role of a captain. Project created by Ariel Piecha, current status: BETA

Sporty © 2016 All rights reserved.

Contact Us Icons designed by Freepik Made with ❤ and Laravel

[Top ↑](#)

Appendix 17 – Research Integrity Certificate

This is to certify that

Ariel Piecha

has successfully completed

Research Integrity (taught courses)

on

10 February 2016