DEVELOPMENT OF A WEB BASED APPLICATION DESIGNED FOR COMMUNICATION BETWEEN UNIVERSITIES SPORT CLUBS

Victor Meriacri
U1768775

School of Computing and Engineering
BSc Information and Communication Technology
University of Huddersfield

July 2020

Abstract

This paper presents research and development of a mobile responsive web application for enhancing the interactions and communication between UK universities sport clubs using modern advancements in Information and Communication Technologies (ICT), open source web libraries and frameworks. The web application helps improving the communication between universities sport clubs by providing the following services: (1) shows and stores data about the sport club, its members and contact details, (2) friendly and easy to follow interface to optimize the organization of friendly games, common trainings or (3) tournaments deployment, (4) advertises social events organized by students into sports, (5) provides marketplace for selling/buying sport items. The web application is developed using several programming languages (PHP, JavaScript, SQL), libraries (Leaflet), CSS pre-processors (SASS) and web frameworks (Laravel). The presented web application has several main advantages: it has a responsive design and works well both on desktop or mobile device; the code is easy to maintain, as it is written using MVC (Model View Controller) method; it is flexible for adding additional web components and libraries; it has been optimized to load fast. Considering all the services provided by the application, students can start using it on a daily basis, thinking of it as a sport social media website/application.

Table of Contents

Chapter 1
Introduction

1.1 Background

Universities are high-level educational institutions in which a lot of people spend their most beautiful and unforgettable life moments. Many factors may affect student life experience, one of them is university sports clubs. Being a sport club member means not only practising sports and staying healthy, it means a lot more. By joining a sport club, a student become part of a whole community which can make his/her student life experience more enjoyable. The social part includes: meeting many new people with same interests, making friends, organizing, or participating in different social or sport events. At the same part, the student can experience a real sport life, which include training sessions, friendly games, competitions, and tournaments. Unfortunately, most of the universities sport clubs are limited by a range of factors and do not communicate with other universities. Lack of communication between universities leads to many opportunities being missed.

My project idea is to produce a web-driven application that can improve the overall level of interactions between universities sport clubs around the UK. The application is going to allow students to register their sport club withing the application and take advantage of a big number of opportunities, which include in-app chat system with other clubs; simplified and automatized process of tournaments organization; simplified process of friendly games organization, which use a request-answer technology; marketplace for selling/buying sport items; advertising system for sport summer camps, tours or events.

1.2 Aim

My aim is to produce a web-driven application for students which are part of their university sport community. The main purpose of the application is to enhance the communication between universities sport teams which will lead to a series of opportunities that can contribute to a better student life experience. First, the application will provide the names and contacts of every sport club member, making it easier to find and contact other universities sport teams. Second, it will provide simple and user-friendly interfaces that will help sport clubs to communicate with each other, and facilitate the process of organizing common training sessions, friendly games, or tournaments. Students interested in sports will be able to look for summer sport camps, sport tours, social events in their area where they can meet people with same passion and socialize. The application will provide a marketplace, where users will be able to sell their sport equipment or buy new or used sport items on reasonable and affordable prices. Every sport club is going to have a unique profile page, where the information about club’s activity and achievements will be displayed for public. This feature can help the club to attract sponsors, it can also help students who want to switch universities with their decision.

1.3 Objectives

I have set the following objectives that I would like to achieve in this project:

1. To research on similar existing web and mobile applications and find out strengths and weaknesses of each.
2. To analyse research data and decide which features can be implemented in my application
3. To research on System Development Life Cycles (SDLCM)
4. To decide on a SDLCM and start on application design
5. To interview students from different sport clubs about the things they would like to see in an application of this kind
6. To research on designing method using personas
7. To build two personas
8. To produce the User Requirements based on interviews and personas
9. To develop the sitemap and task flow diagrams
10. To design application wireframes
11. To find out the key changes in design
12. To research on process of application prototyping
13. To develop first application prototype
14. To elaborate user testing plan
15. To test the prototype on users and gather the data
16. To do a deep analysis of data gathered and make conclusions
17. To develop the final application prototype
18. To research on web technologies, frameworks, libraries, and methodologies
19. To decide which modern web development technologies to use
20. To design the database
21. To develop the actual application
22. To evaluate the application
23. To set future goals

1.4 Motivation

For many students, as for me, university sport club became part of the life. People would always strive to improve their life quality, by simplifying some processes, inventing new things that would do actions for them, etc. I am not an exception, I know that I can build something useful for the students like me, and this project served as a big motivation to do so. Being part of volleyball club for 3 years, and vice-president of the club in my final year, I realized how many opportunities we have lost due to lack of communication with other universities teams. Too many actions were required to arrange and organize a banal friendly game or common training session, and it was very time-consuming. The problem was in finding the right contacts, and in the way of contact itself. The stages of a friendly game organization process between two universities would be: (1) one part finds the contacts of other part representative, (2) get in touch with each other, (3) request a friendly game, (4) discuss the available dates and times, (5) discuss and decide location, (6) meet and play.
The most time-consuming stages are 1, 4, the time needed for first stage can be reduced if the clubs contacted each other before, but the stage number 4 cannot be ignored. The application

would help in simplifying and almost automatization of the stages 1, 2, 4 and 5. The application will give a huge boost to sport students' life quality, by helping them not only participate in different competitions, games, training, but also joining sport summer camps, social events, which can contribute to growth of sport professionalism and social skills of the students.

1.5 Relation to course

This project is a challenge for me, a challenge to my professional and technical skills gained during these 3 years of studying Information and Communication Technologies. The development of the application incorporates a lot of difficult tasks, such as, database design, front-end and back-end development, UX design. Luckily, I have studied everything about applications design and development during my course, now it's the best time for me to demonstrate my abilities.

Chapter 2
Design

2.1 SDLCM

Software development life cycle (SDLC) is a method by which the software can be developed in a systematic manner and which will increase the probability of completing the software project within the time deadline and maintaining the quality of the software product as per the standard [1]. There are various SDLC models widely used for developing software, it is important to decide on which SDLCM to use according to the type and size of project.

a. Waterfall Model

Waterfall model was proposed by Royce in 1970 which is a linear sequential software development life cycle (SDLC) model. The various phases followed are requirements analysis, design, coding, testing and implementation in such a manner that the phase once over is not repeated again and the development does not move to next phase until and unless the previous phase is completely completed [1]. The waterfall model's advantages are simplicity and low costs development.

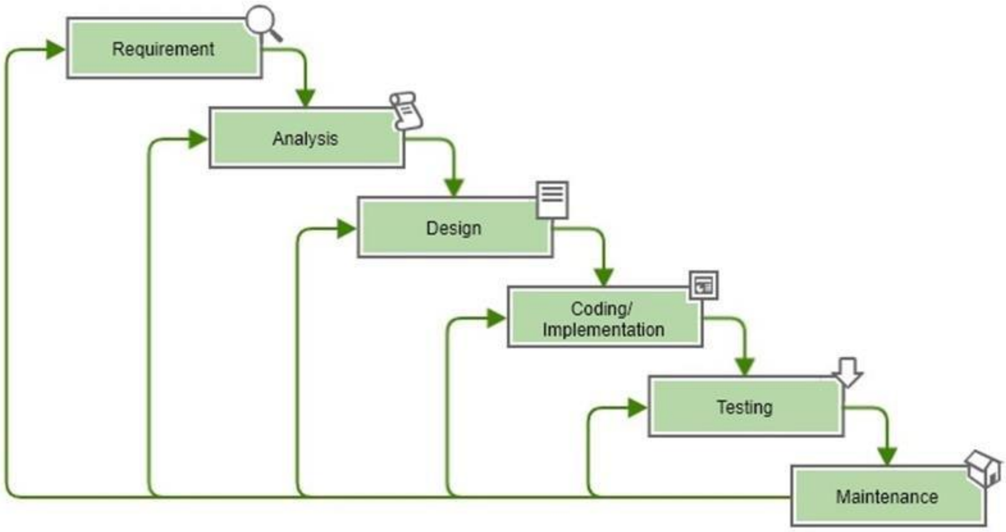


Figure 2.1 Waterfall SDLC

b. Spiral Model

In response to the weaknesses and failures of the Waterfall SDLC Model, many new models were developed that add some form of iteration to the software development process. In the Spiral SDLC Model as in figure 1.2, the development team starts with a small set of requirements and goes through each development phase (except Installation and Maintenance) for those set of requirements [14]. Based on lesson learned from the initial iteration, the development team adds functionality for additional requirements in ever-increasing "spirals" until the application is ready for the Installation and Maintenance phase.[1] The disadvantages of the spiral model against waterfall model are: high development costs, and increased difficulty.

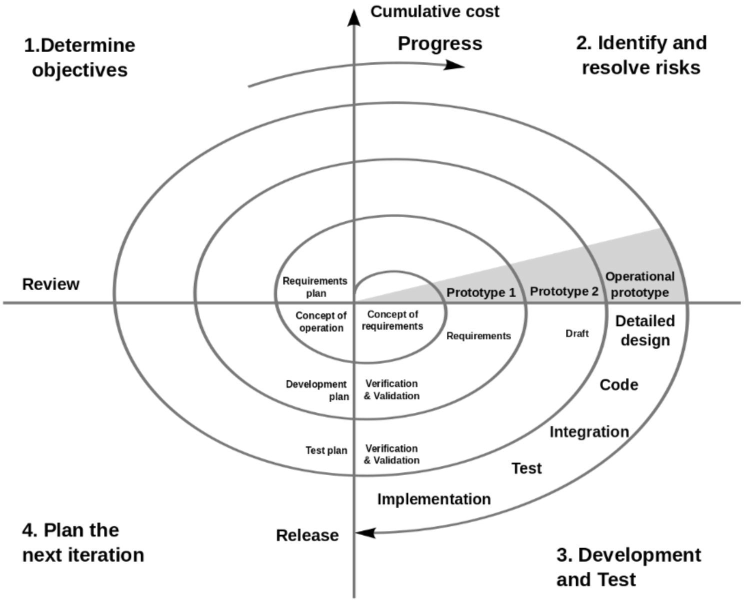


Figure 2.2 Spiral Model

c. V-Shaped Model

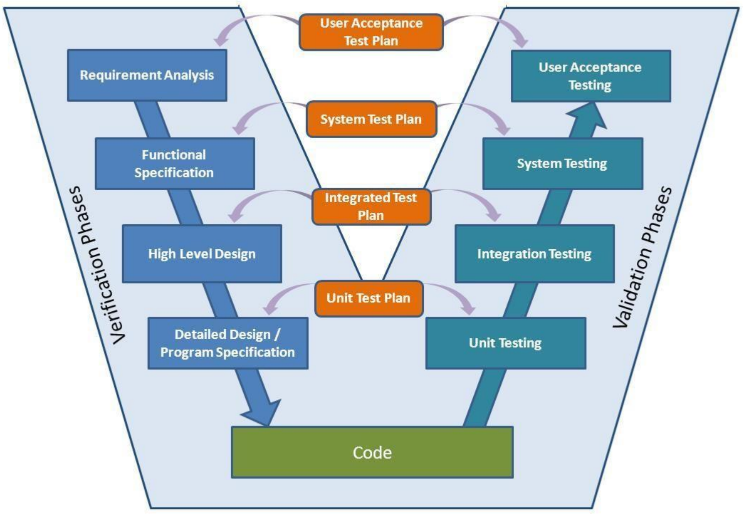
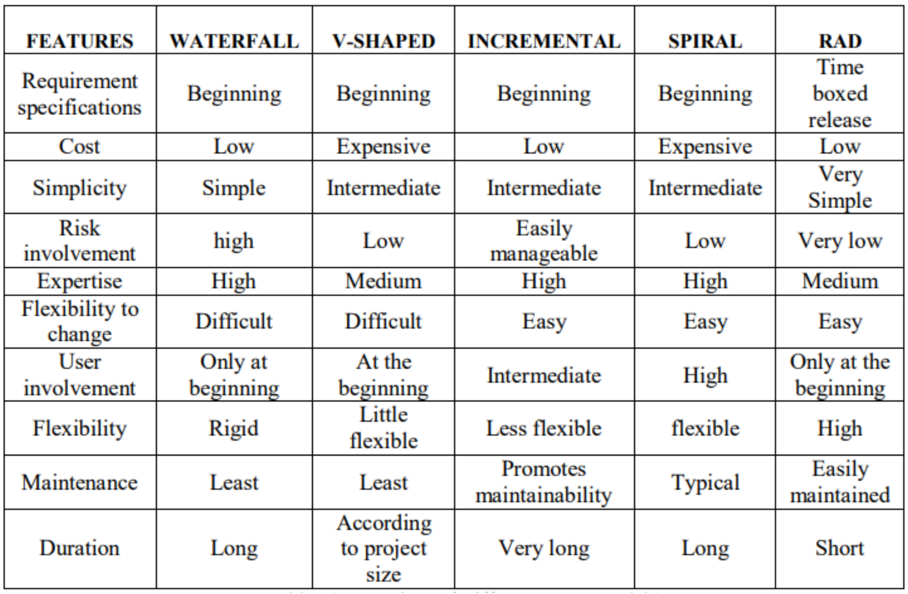


Figure 2.3 V-Shaped Model

A variant of the Waterfall that emphasizes the verification and validation of the product, testing of the product is planned in parallel with a corresponding phase of development.

There are many SDLC models, each has its own advantages and disadvantages. When deciding upon which one to use, it is important to understand the size of the project and the developers skills set, as well as the time limits, development costs etc.



FEATURES	WATERFALL	V-SHAPED	INCREMENTAL	SPIRAL	RAD
Requirement specifications	Beginning	Beginning	Beginning	Beginning	Time boxed release
Cost	Low	Expensive	Low	Expensive	Low
Simplicity	Simple	Intermediate	Intermediate	Intermediate	Very Simple
Risk involvement	high	Low	Easily manageable	Low	Very low
Expertise	High	Medium	High	High	Medium
Flexibility to change	Difficult	Difficult	Easy	Easy	Easy
User involvement	Only at beginning	At the beginning	Intermediate	High	Only at the beginning
Flexibility	Rigid	Little flexible	Less flexible	flexible	High
Maintenance	Least	Least	Promotes maintainability	Typical	Easily maintained
Duration	Long	According to project size	Very long	Long	Short

Figure 2.4 Comparison of Different SDLC Models

Decision

My decision was based on several factors. First, I had to choose a system development life cycle model that is not too expensive. Second, the SDLC model should not be difficult to use. Lastly, the model should be optimized to working alone and the duration should not exceed 2 months.
I have decided on customizing my own SDLCM based on Waterfall model. Waterfall SDLC model is perfect for low cost, medium-sized types of projects, which suits my project. It is also simple, and the duration period was shortened by me using some sort of iterations during the process. The customization of Waterfall model included: (1) one more testing phase was added right after design stage, (2) iteration approach used during stage number 3 (Design) and stage number 5 (Testing). Overall, the resulted SDLC model should have the same advantages of Waterfall model but reduce the development risks due to implementing iteration approach.

2.2 Research on Existing Systems

My strategy was to research on existing application, gather data about their pluses and minuses. By doing a heuristic analysis over the following applications, I was able to find out which features are worth implementing in my application as well.

a. Data Gathering

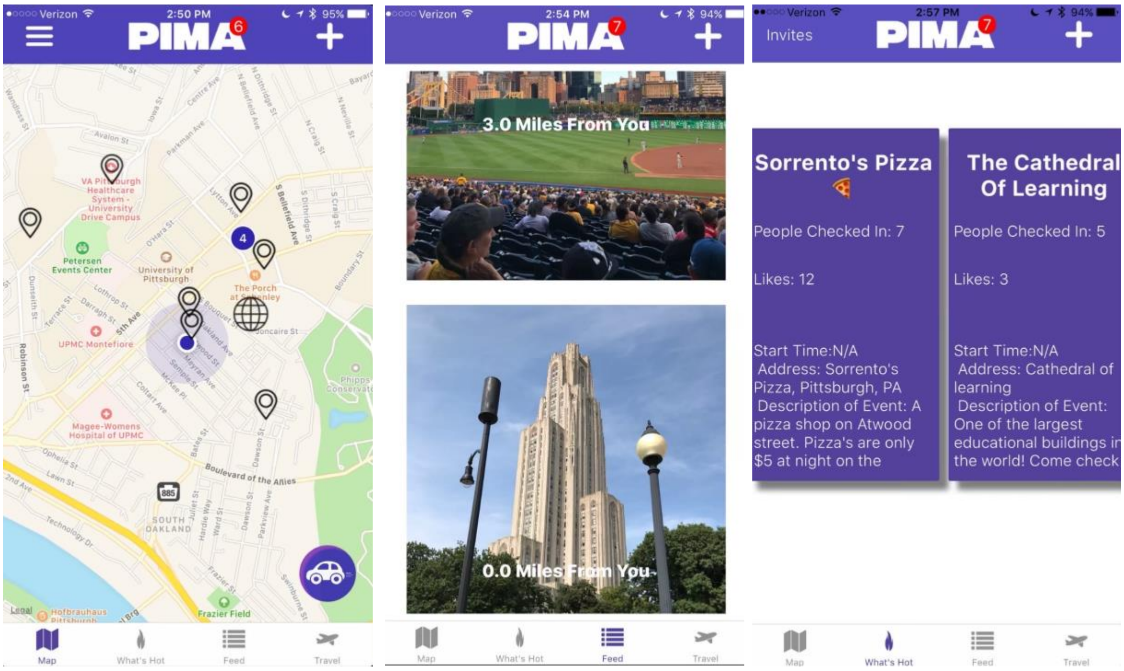
The application I am going to develop, which will be mentioned later under the name of studSports, helps its users achieve five different goals:

1. Browsing or advertising events (Events)
2. Browsing sport clubs and see information about them (Sport Clubs)
3. Viewing information about sport tournaments (Tournaments)
4. Buy or sell sport equipment/items (Trading)
5. View information about summer sport camps and tours or advertise them (Advertising)

As there are no such existing applications that can be that complex, I have decided to research on separate apps for the above goals. In this way, I have chosen the following applications for further research and analysis:

1. Events and Advertising: Pima
2. Sport Clubs and Tournaments: BUCS Play
3. Trading: Facebook Marketplace

Pima: People, Places, Parties in My Area is an interactive mobile application used for connecting people within the community around them. Using Pima users can create their own events that are going to be shown on the map inside the application, users also are able to check-in to existing events.



Sorrento's Pizza

People Checked In: 7

Likes: 12

Start Time:N/A
Address: Sorrento's Pizza, Pittsburgh, PA
Description of Event: A pizza shop on Atwood street. Pizza's are only $5 at night on the

The Cathedral Of Learning

People Checked In: 5

Likes: 3

Start Time:N/A
Address: Cathedral of learning
Description of Event: One of the largest educational buildings i the world! Come check

Figure 2.5 Pima Application

BUCS Play is a mobile application developed for British university and college sport competitions. Within the application, users can manage their team, view fixtures, check results and league tables. The application can be used only by users that are part of their educational institutions sport teams which were previously registered to play in BUCS League.

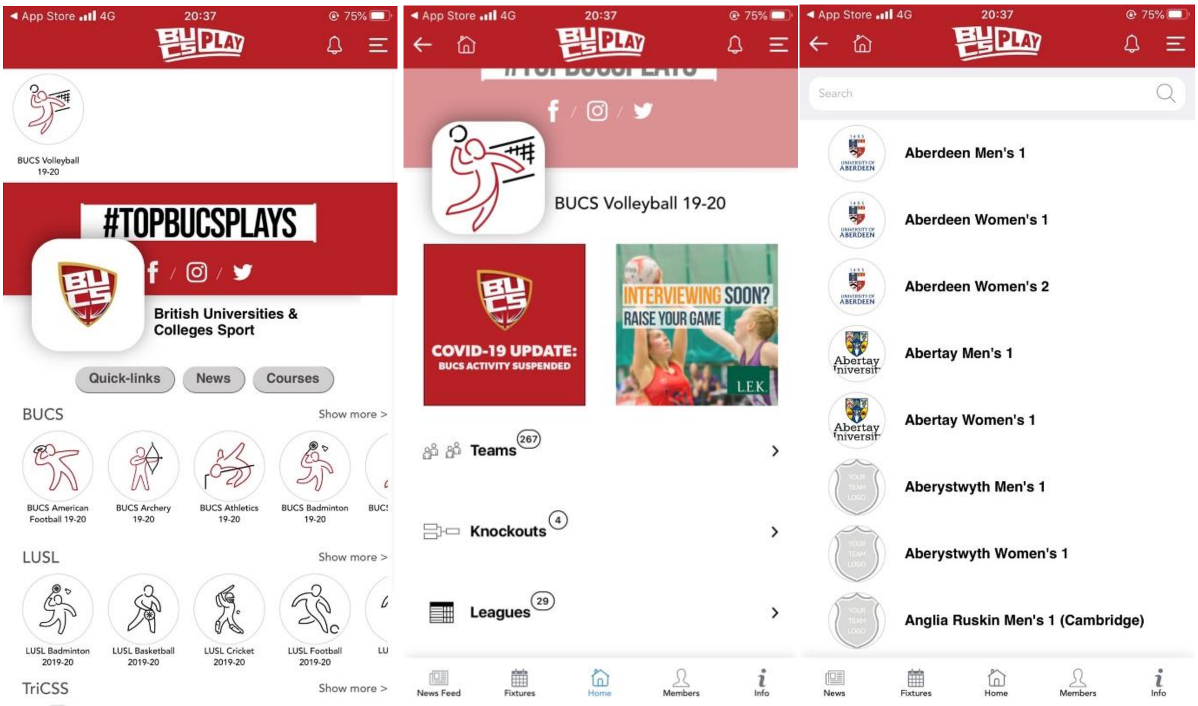
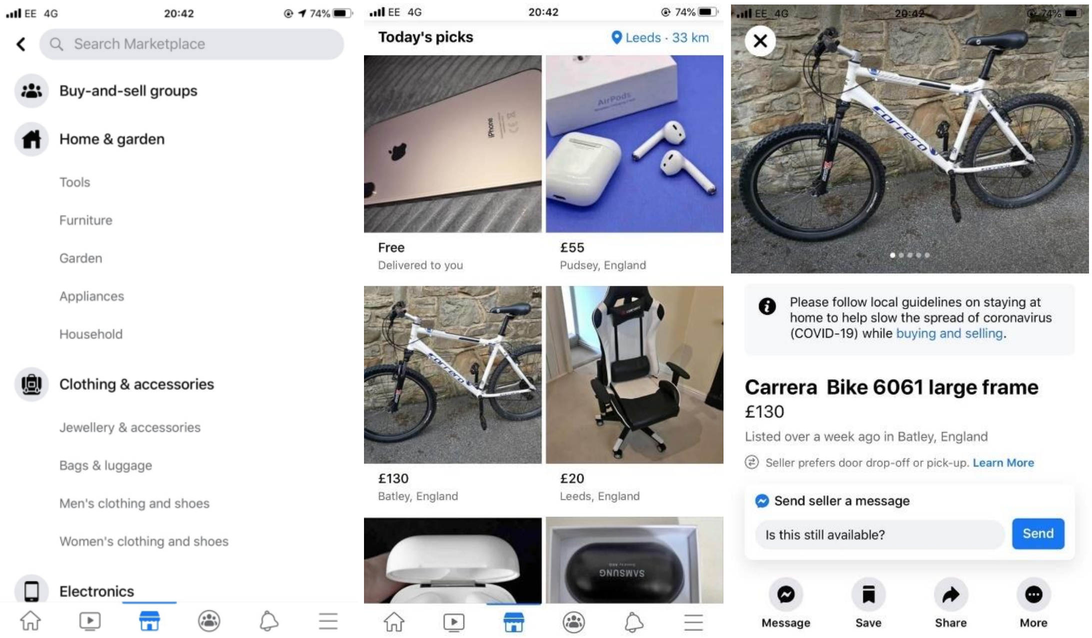


Figure 2.6 BUCS Play Application

Facebook Marketplace represents a section in the official Facebook application where users can list their items for sale, as well as communicate with buyers and sellers through messenger.



Please follow local guidelines on staying at home to help slow the spread of coronavirus (COVID-19) while buying and selling.

Carrera Bike 6061 large frame
£130
Listed over a week ago in Batley, England
Seller prefers door drop-off or pick-up. Learn More
Send seller a message
Is this still available?
Send

Figure 2.7 Facebook Marketplace Section

b. Key findings from data gathering

Heuristic analysis based on Nielsen's Usability Heuristics (check appendix [1])
Each application was analysed against each heuristic, one at a time. As a design was first assessed for 'user control and freedom', if it contained a feature that supported this heuristic it scored 1 point, if it contained two or more features it scored 2 points. If no features supported that heuristic the score was 0. The analysis was carried out for each of the heuristics in this way.

Total Heuristic Score by Application

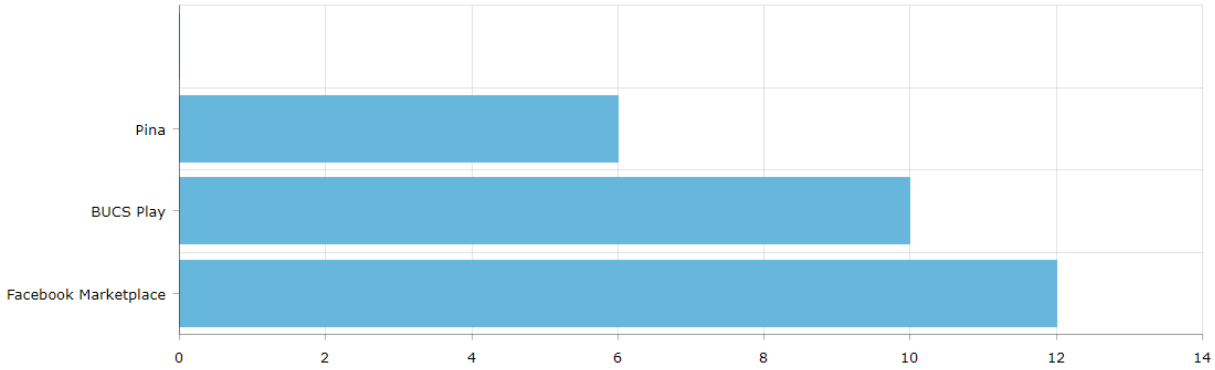


Table 2.1 Heuristic Score of Applications

After applying heuristic analysis on ***Pima*** application, I have figured out that some of the principles, such as, "recognition rather than call" and "help and documentation" are using a poor approach, so it does not fulfil the needs of every category of users. First, the dialogs that can seem easy at the first view, turn out to be difficult to follow for unexperienced users, as it requires memorising some things. In addition, saved or created events cannot be accessed offline, which can cause problems.
BUCS Play and ***Facebook Marketplace*** design was much better. Some of the best points are: remembering the users choice, a good use of filtering search pattern, the accessibility features are well-designed and eliminates the risk of experience the "dead-ends", and both applications provide a simple and user friendly design of the bottom navigation bar. The missing feature include a more detailed team profile (BUCS Play) and a price-range filter (Facebook marketplace).

By analysing each application, I have found out that each of them has their plusses and minuses. Every application does its main function well, but there is room for improvement. I figured out several interesting features I would like to implement in my application:

Events and Advertising:

- Application provides a map with all the events listed on it
- Users can add their own event to the map
- Users can advertise your event in feed list
- Users can check-in to events
- "What's Hot" list of events

Sport Clubs and Tournaments:

- Categorize the clubs by sport type
- Access the results of the team
- View team members
- View results of tournaments
- Check the tournament brackets

Trading:

- Filter the items by different category
- Contact seller button
- Location of listing is marked on map
- Friendly and simple interface when creating a listing
- Album paginated photo viewing

2.3 User Requirements

I have used several techniques to get the user requirements, personas, and interviews (surveys). It was important to realise that without real people participating in an app design and development, the application can easily fail. A good and useful application is the one which considers people's thoughts, ideas and wishes. I build the user requirements not only based on my own thoughts, I did surveys to interrogate the users, I also built two personas based on my real students.

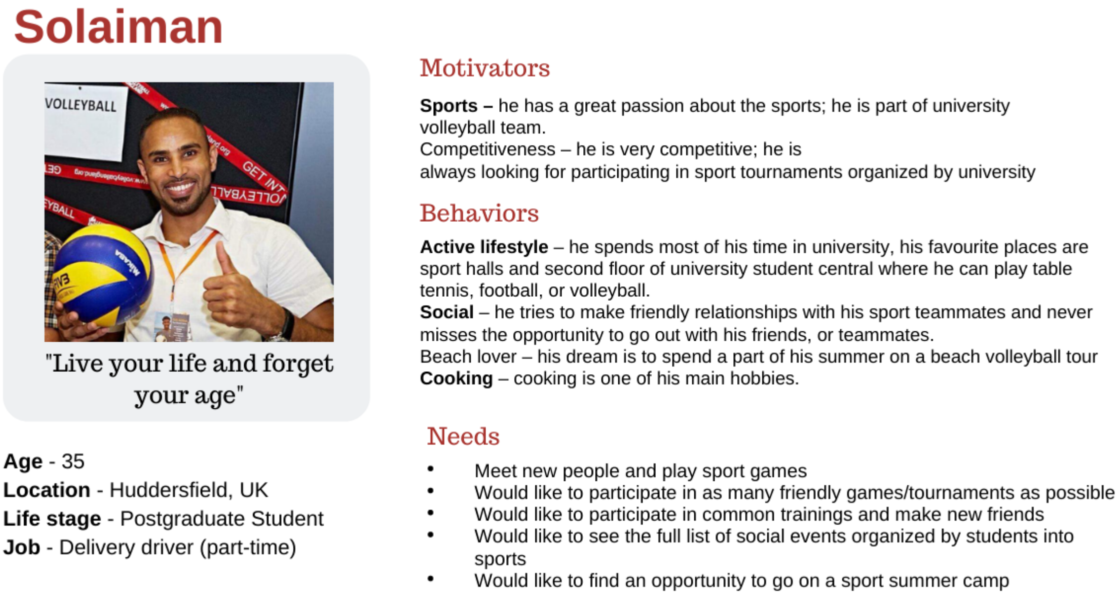
a. Interviews

I have developed a number of questions based on sport life in universities, needs, social life within sport club/team. I have decided on interrogating the people directly, via Facebook messenger chat, or voice calls. I made this decision based on two factors. First, I have contacts of many students which are part of various sport teams, these people know me and are willing to help. Second, some people use to drop online surveys without finishing them, or they think they are too busy to do it. Asking them in private, made them feel the importance of this action for me, and most of them accepted to help me in the same day. Even though, it took more time, the result satisfied me (the result is discussed in the *table 2.2*).

b. Personas

The use of personas as a method for communicating user requirements in collaborative design environments is well established. Personas are characters, which I created based upon my research in order to represent the different user types that might use my application in a similar way. Creating personas helped me understand my users' needs, experiences, behaviours, and goals. Personas make the design task at hand less complex, they guide your ideation processes, and they can help you to achieve the goal of creating a good user experience for your target user group. [2] Using this design approach I acknowledged that different people have different needs and expectations. I built two personas, which have different motivation and different needs.

Persona 1 – Solaiman



Solaiman

"Live your life and forget your age"

Age - 35
Location - Huddersfield, UK
Life stage - Postgraduate Student
Job - Delivery driver (part-time)

Motivators

Sports – he has a great passion about the sports; he is part of university volleyball team.
Competitiveness – he is very competitive; he is always looking for participating in sport tournaments organized by university

Behaviors

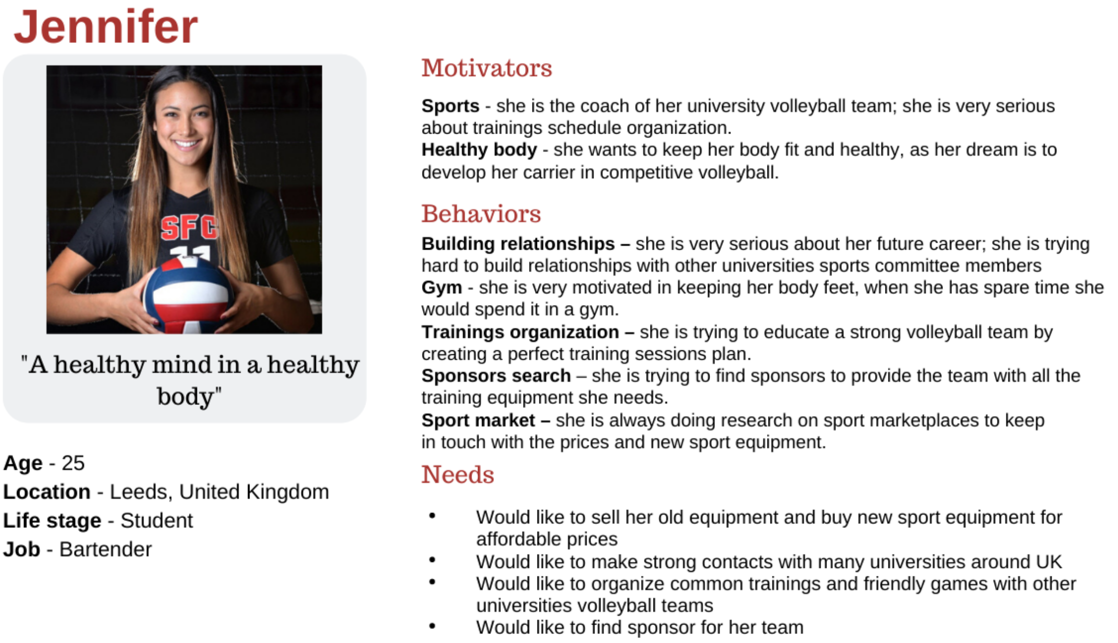
Active lifestyle – he spends most of his time in university, his favourite places are sport halls and second floor of university student central where he can play table tennis, football, or volleyball.
Social – he tries to make friendly relationships with his sport teammates and never misses the opportunity to go out with his friends, or teammates.
Beach lover – his dream is to spend a part of his summer on a beach volleyball tour
Cooking – cooking is one of his main hobbies.

Needs

- Meet new people and play sport games
- Would like to participate in as many friendly games/tournaments as possible
- Would like to participate in common trainings and make new friends
- Would like to see the full list of social events organized by students into sports
- Would like to find an opportunity to go on a sport summer camp

Figure 2.8 Persona 1

Persona 2 – Jennifer.



Jennifer

"A healthy mind in a healthy body"

Age - 25
Location - Leeds, United Kingdom
Life stage - Student
Job - Bartender

Motivators

Sports - she is the coach of her university volleyball team; she is very serious about trainings schedule organization.
Healthy body - she wants to keep her body fit and healthy, as her dream is to develop her carrier in competitive volleyball.

Behaviors

Building relationships – she is very serious about her future career; she is trying hard to build relationships with other universities sports committee members
Gym - she is very motivated in keeping her body feet, when she has spare time she would spend it in a gym.
Trainings organization – she is trying to educate a strong volleyball team by creating a perfect training sessions plan.
Sponsors search – she is trying to find sponsors to provide the team with all the training equipment she needs.
Sport market – she is always doing research on sport marketplaces to keep in touch with the prices and new sport equipment.

Needs

- Would like to sell her old equipment and buy new sport equipment for affordable prices
- Would like to make strong contacts with many universities around UK
- Would like to organize common trainings and friendly games with other universities volleyball teams
- Would like to find sponsor for her team

Figure 2.9 Persona 2

c. User Requirements Table

Based on personas and interviews, I have developed the following set of user requirements.

Claim/findings	User Requirements	Rationale
Survey 1 30% would use a sport application to buy or sell sport equipment, 70% would use it to search for sport events in the area, 15% would use it to advertise their own events	Users should be able to search for sport equipment listings Users should be able to buy items on marketplace Users should be able to create their own listings and sell items on marketplace Users should be able to look for sport and social events happening in their country Users should be able to place an advertisement over their own sport/social events	Users should be able to sell their old sport equipment (Jennifer) Users should be able to buy new or used sport equipment (Jennifer) Users should be able to see the full list of events organized by students into sports (Solaiman)
Survey 2 80% would like to find other sport teams contacts easier, 60% would use the application to look for a common training partner, 50% would use it to organize friendly games with other universities	Information about the university sport club should display the contact information for public use Users should be able to contact other club using the application Application should provide an easy step-by-step interface to facilitate the organization of friendly games and matches	Users should be able find contacts of sport clubs commitment members (Jennifer) Users should be able to request a common training session with other university team (Jennifer) Users should be able to organize friendly games or matches with other universities using the application (Solaiman, Jennifer)
Survey 3 40% would like to see the information about the activity of different university clubs, 70% would like to share their sport club activity and achievements	Application should provide a unique profile for every sport club Profile should store and show the information about club, their activity, and achievements Users should be able to search for a club and see the information about it Users should be able to share achievements and activity of their club	Users should be able to share activity and achievements of their club (thus attracting new members and sponsors, Jennifer)

Table 2.2 User requirements

2.4 Sitemap and Task Flow

a. Task flow diagram

The following task flow diagram is related to the users that are interested in organizing common training session with other universities sport clubs. It is related to Jennifer, as her main goal is to organize training sessions that can improve her team performance.

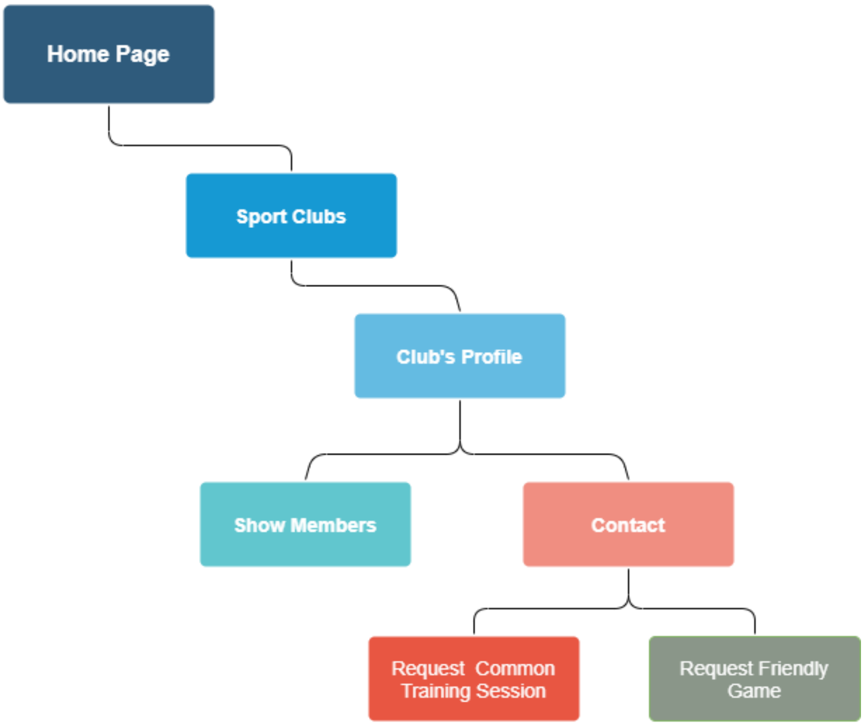


Figure 2.10 Task Flow diagram

Home Page - On this page users will be able to choose the main task they would like to perform (Browse Sport Clubs, Browse Events, View Tournaments, View available Camps and Tours, View Marketplace). Here users can also access the advance search feature.

Sport Clubs - On this page, a list of all sport clubs registered in the application will be displayed. Users will be able to filter the results by several categories: university, city, and sport type (ex. rugby, volleyball, swimming).

Club’s Profile Page - is going to show the information about a specific sport club. The information displayed will include: club name and description, university, club’s profile picture, list of members, contact page and club’s activity and achievements.

Show Members - page is going to show every member that is part of the club, including links to their profile, contact details and position within the club (ex. member, president etc.)

Contact Page - will provide two options, request common training session or request friendly game.

Request pages are pretty much identical, with some minor design changes. Request training session page is going to display a calendar where the available and unavailable date slots of the club will be displayed, the user should select from available slots to proceed with the claim. The claim is considered completed when the date slots are selected, and the additional information, such as, training location, number of members allowed, training plan, is being completed.

b. Sitemap

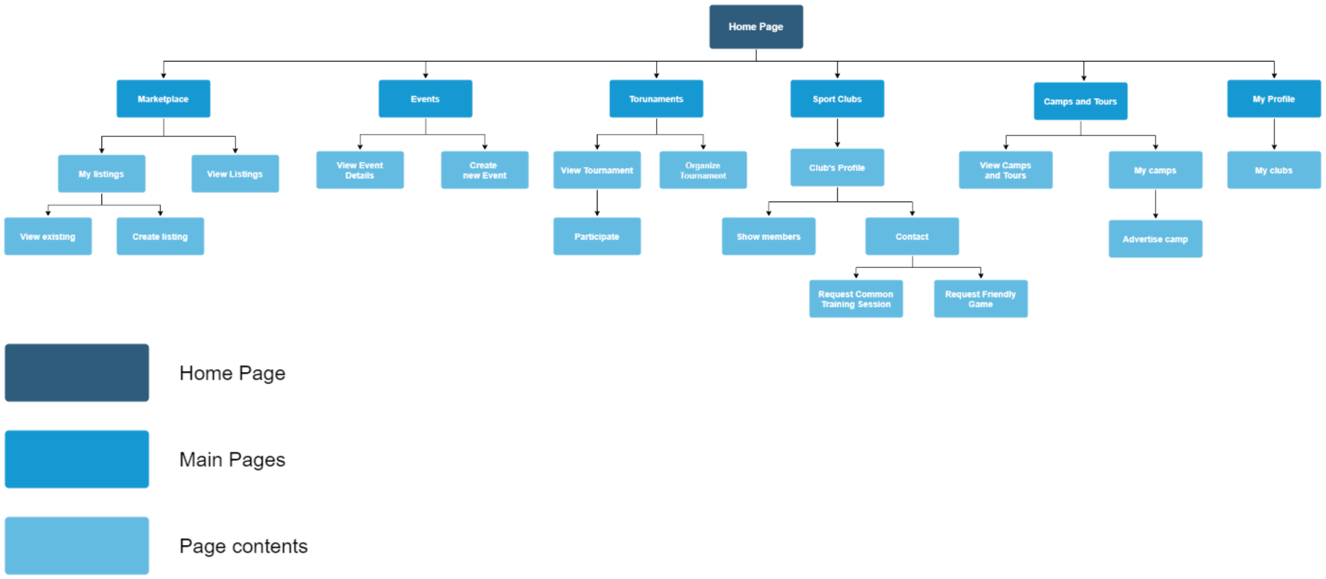
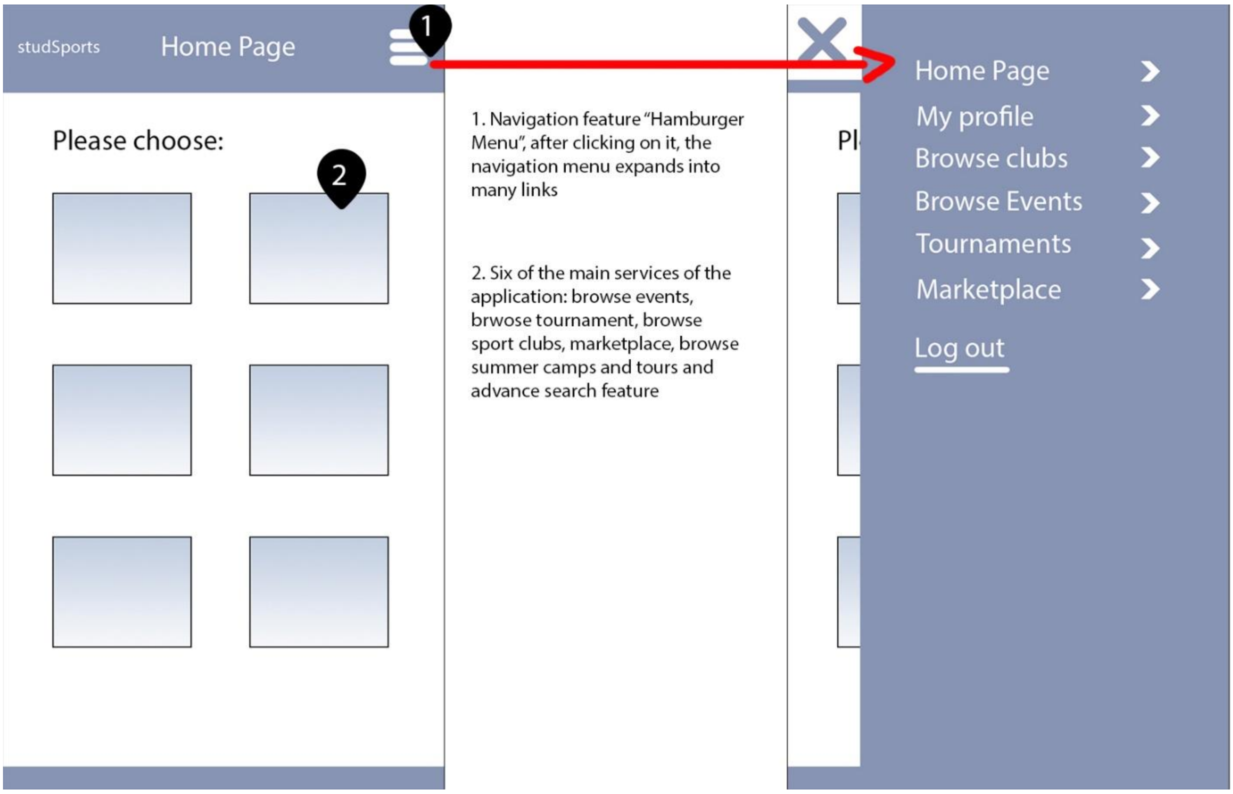


Figure 2.11 Sitemap

I structured the content in this way to make the use of application as simple as possible, so that even unexperienced users will be able to understand and follow the way to their goals in using it.

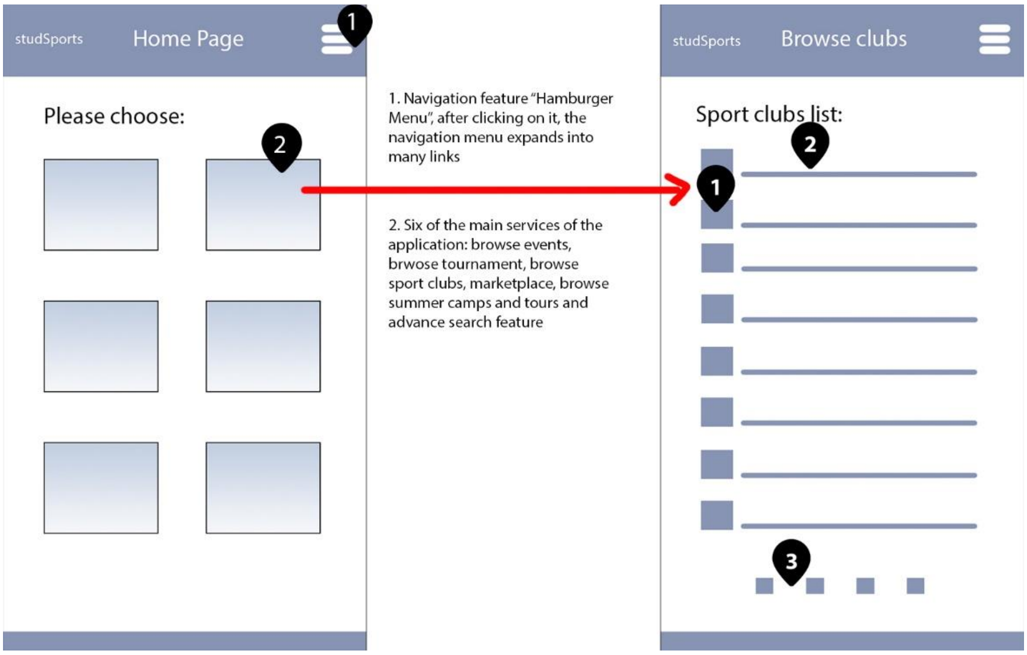
c. Wireframes



1. Navigation feature "Hamburger Menu", after clicking on it, the navigation menu expands into many links

2. Six of the main services of the application: browse events, brwose tournament, browse sport clubs, marketplace, browse summer camps and tours and advance search feature

Figure 2.12 Home Page and Navigation Menu Wireframes



1. Navigation feature "Hamburger Menu", after clicking on it, the navigation menu expands into many links

2. Six of the main services of the application: browse events, brwose tournament, browse sport clubs, marketplace, browse summer camps and tours and advance search feature

Figure 2.13 Home and Browse Clubs Page wireframes

Browse clubs Page shows the list of sport clubs registered within the application. Marker (1) represents the logo of the sport club, marker (2) is the name of club. Marker (3) represents the pagination feature, so that only 10 results can be displayed at one time.

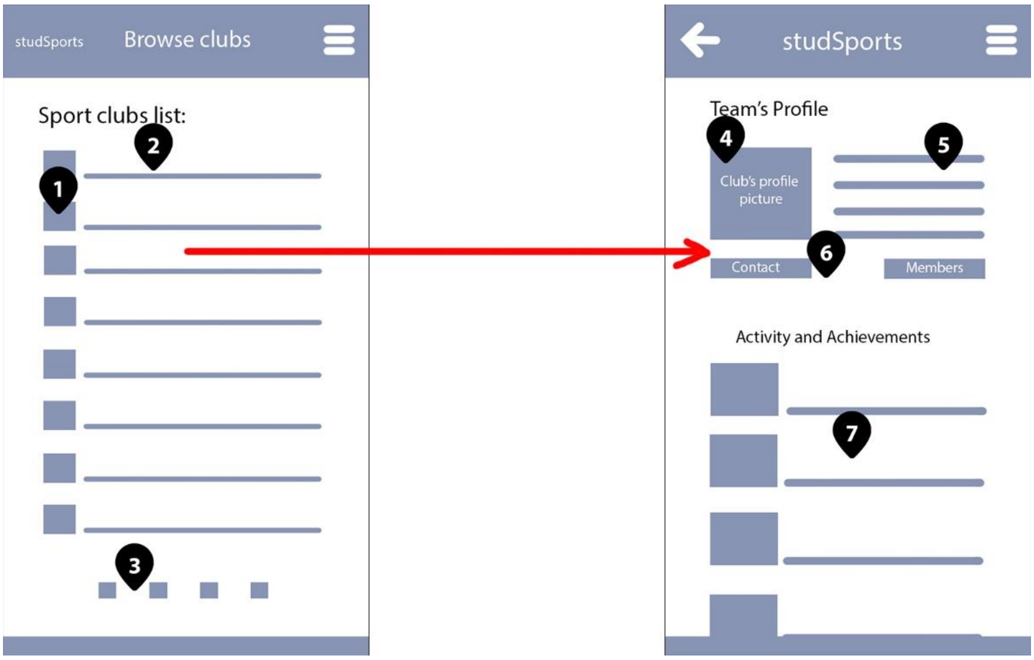


Figure 2.14 Browse Clubs and Club Profile wireframes

A team profile page consists of (4) profile logo/picture, (5) name of the club, description, (6) button links to contact and members pages, (7) recent activity and achievements of the club. There is a "go back" arrow in the heading of the page, as well as the navigation "hamburger" menu, to prevent dead ends while using the application.

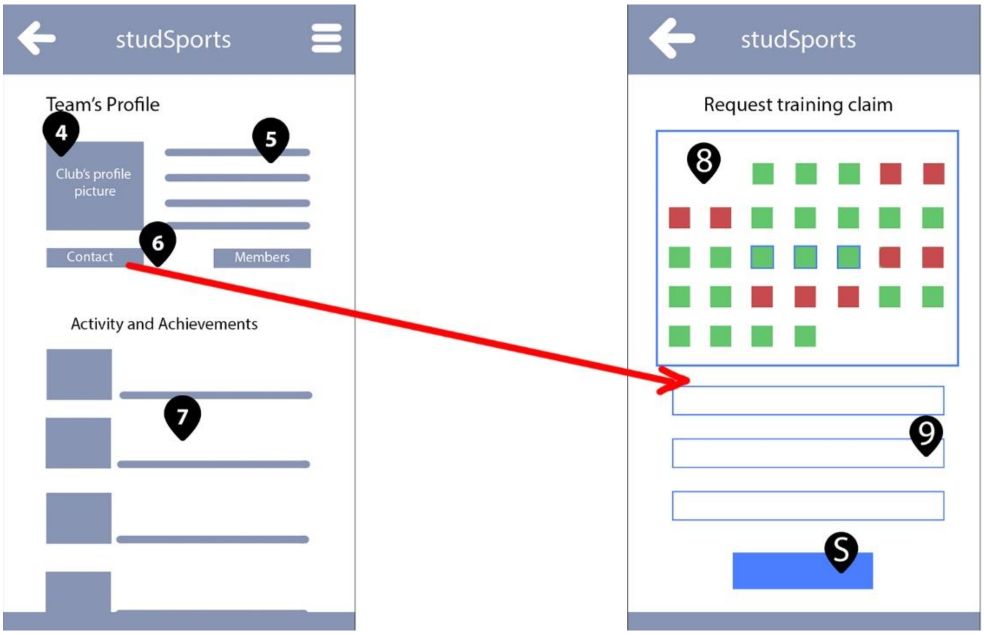


Figure 2.15 Club's Profile and Request page wireframes

The connection between profile page and request training page is done via (6) contact button. The request training claim page contains (8) a calendar which shows the available and unavailable dates in which the team can join the training; (9) text fields are provided to discuss details (location, training session plan, time etc), and (S) submit button.

The above wireframes display the "route" of requesting a common training, wireframes are linked to my sitemap through the home page and the navigation "hamburger" menu, it shows the task flow way and the "escape" routes and buttons. It has several accessibility features that makes it easier for the user to follow the logic. I have tried to make it as compact as possible to avoid extra development costs.

Key design changes:

- Real-time updates on requests: users should be able to request a common training session with a sport club, and track the updates in real time, to see rather than the request is approved or denied, or any changes are suggested by the other part
- Notification system: users should be able to receive notifications about various events in the application and email
- Auction and bids trading feature: users should be able place bids on existing listings
- Club's available timeslots: Users should be able to provide application with the information about when the time their club is free
- Application should provide users with date and timeslots when the other part is free, to make the organization process easier
- Activity of the club and achievements should be displayed on club's profile
- Permission security, only commitment members (user's with special permission level) can request trainings and friendly matches to other clubs.
- Tournaments have their own page now, and can be organized only by club president or vice presidents
- System of invitations to tournaments
- Chat with other members inside the application

2.5 Database Design

Based on User Requirements I have designed the database using draw.io tool.
A class diagram showing relations (Figure 2.16) and a detailed physical diagram showing pivot tables and data type of every field inside the tables (Figure 2.17).

a. Class diagram

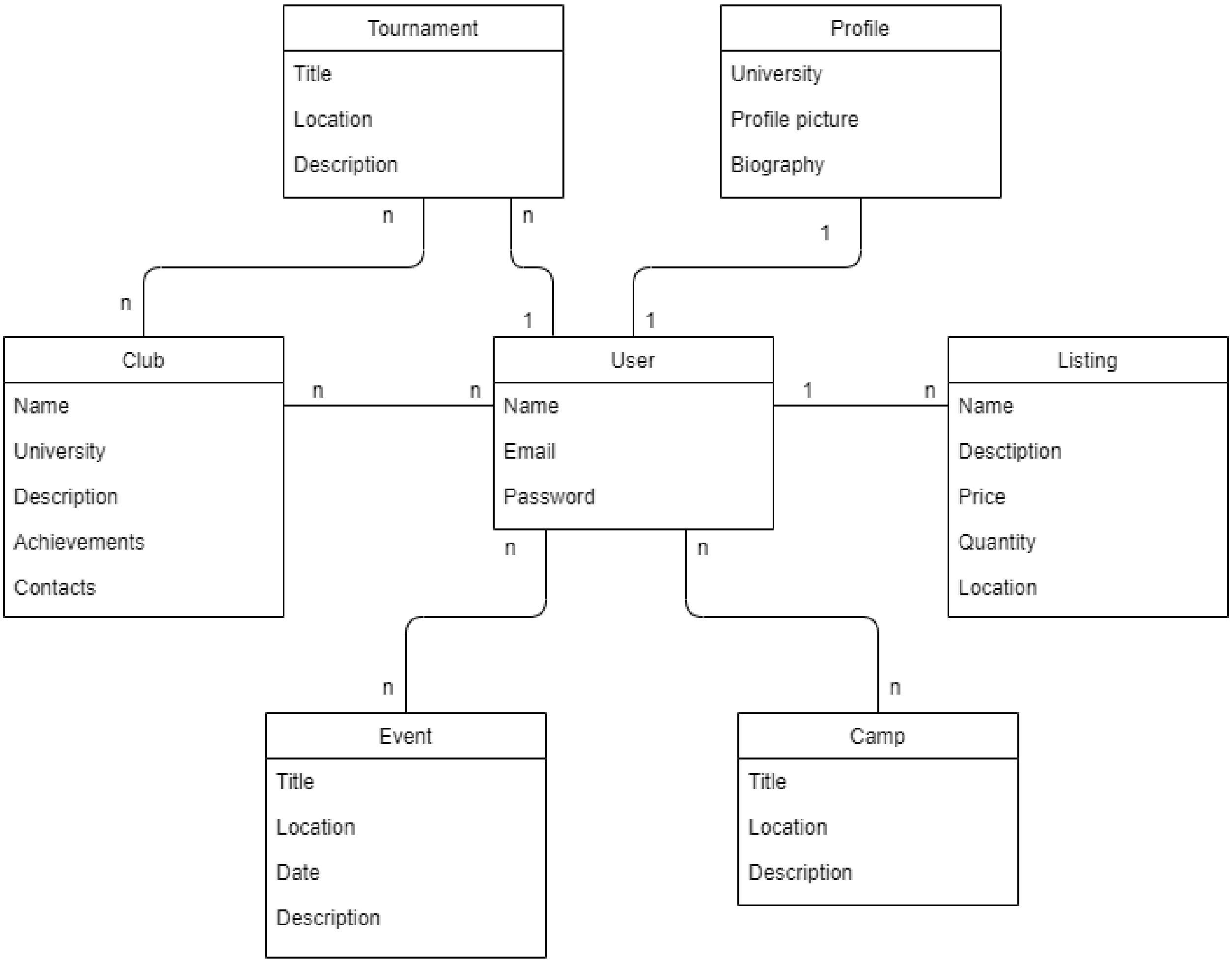


Figure 2.16 Database Class Diagram

b. Physical diagram

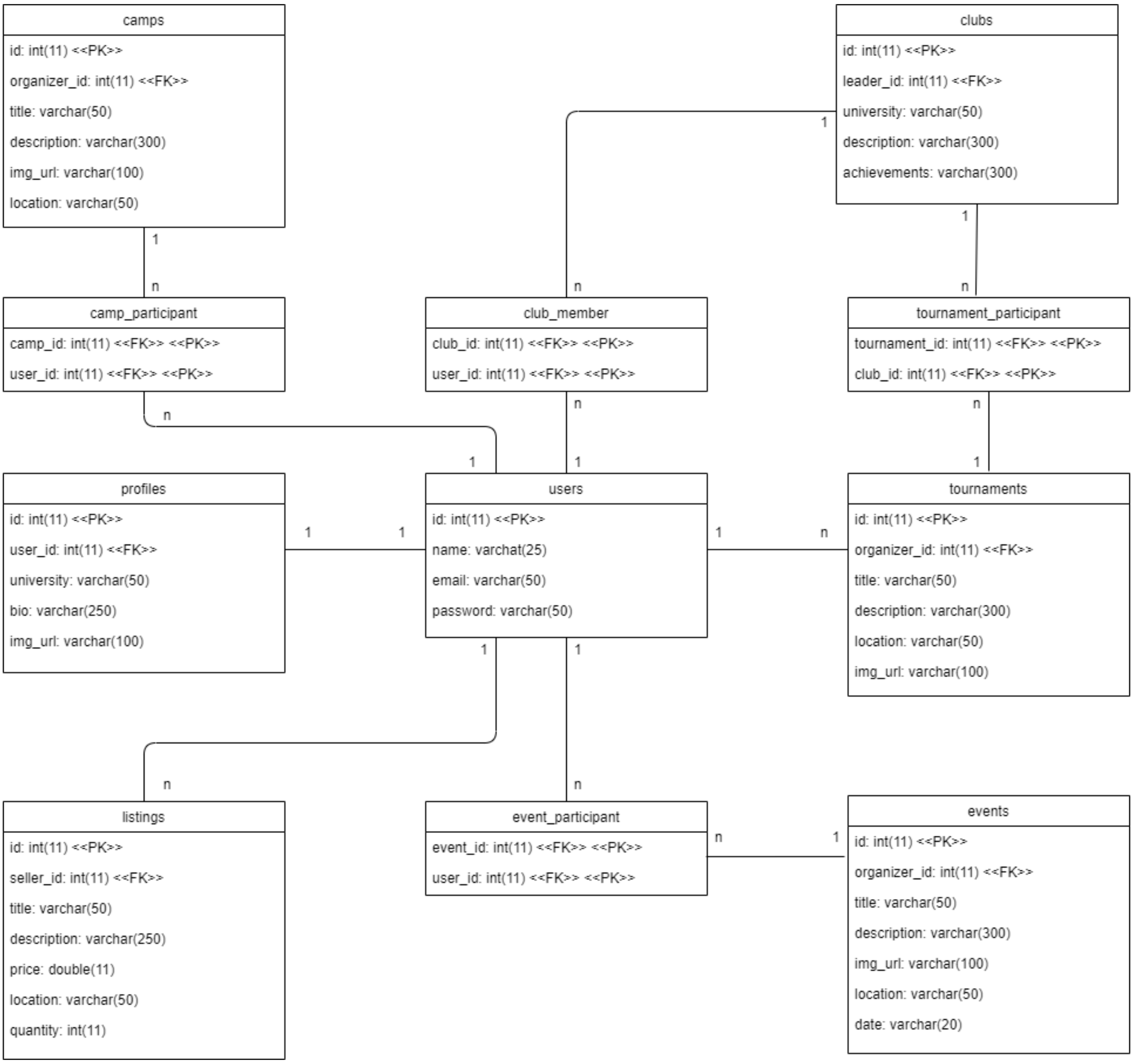


Figure 2.17 Database Physical Diagram

Chapter 3

Prototyping

In this chapter I developed studSports application prototype. I used the pre-defined user requirements, task flow diagrams and wireframes to produce a high-fidelity prototype.

3.1 Conceptual Model

StudSports application is intended to help students with managing their social life and sport activities. The app creates an impression of a reliable friend, that can help you anywhere, at any time. It can share information, help with tracking social events in the area, and keep its users updated with information regarding their sport clubs. StudSports doesn't dictate, but it makes users want to follow it. To create a friendly atmosphere within the application, I have used vector images rather than raster images, in this way users can feel more relaxed and detached from real world while using the application. It creates an atmosphere of a cartoon, where everything is simple and doesn't require any deep intellectual work to understand the meaning and actions flow. StudSports's design is aimed to be understood by people of different age categories, I did my best to keep the design simple and user-friendly. What is more, I focused a lot on navigation system, I have added friendly tips that pop-up every specific amount of time, helping users to get used to application faster. As a true friend, StudSports makes users feel comforted while using it, StudSports provides support not only in carrying out users' personal needs but also helps users to "get along" with the application itself (navigation tips & help centre).

3.2 Paper Prototyping

Before starting the actual studSports app prototyping, I did a deep analysis of the sitemap, task flow diagrams and the wireframes I have designed before. It helped me to understand that one of the best prototyping options is the high-fidelity one, which I firmly chose. This type of prototyping has several big advantages when it comes to user testing, as the high-fidelity prototype looks more like the final product than a low fidelity version, users can carry out their tests more effectively. Most of the users that test my application are using their phones on daily basis, it became a common thing for them to download and run new applications. When providing them with a high-fidelity prototype they were thinking of it as a real application, which sometimes may be dangerous, but not in my case. Using MarvelApp, I have developed the prototypes based on user requirements, wireframes and the task flow diagrams. I chose the "vertical" prototyping compromise and developed prototypes for the request a training session task flow, providing a good amount of details.

The provided bellow prototypes screen shots show the transformation from wireframes to studSports prototype pages.

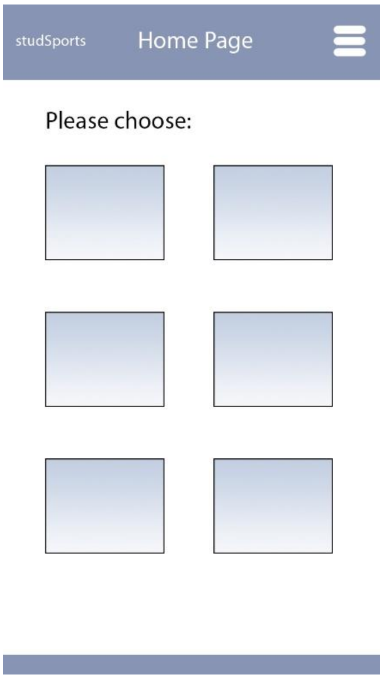


Figure 3.1 Home Page Wireframe

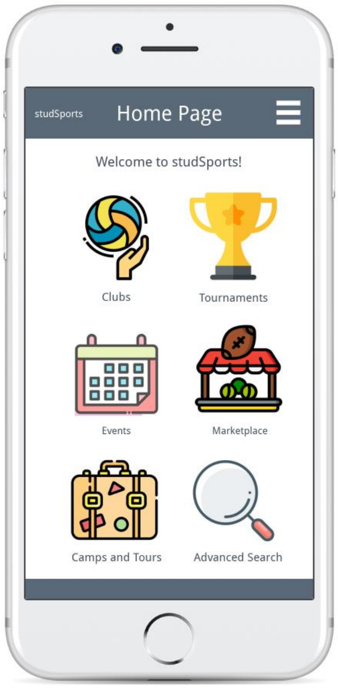


Figure 3.2 Home Page Prototype

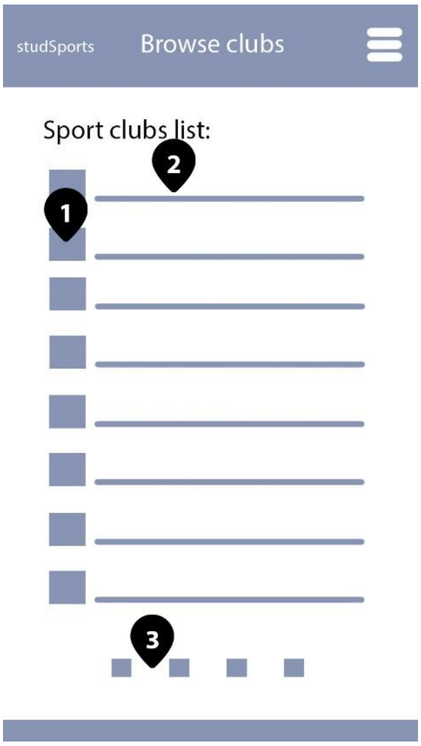


Figure 3.3 Browse Clubs Wireframe

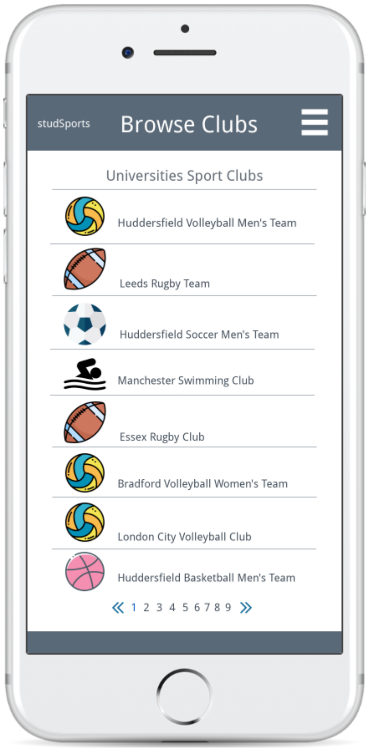

studSports
Browse Clubs
Universities Sport Clubs
Huddersfield Volleyball Men's Team
Leeds Rugby Team
Huddersfield Soccer Men's Team
Manchester Swimming Club
Essex Rugby Club
Bradford Volleyball Women's Team
London City Volleyball Club
Huddersfield Basketball Men's Team
« 1 2 3 4 5 6 7 8 9 »

Figure 3.4 Browse Clubs Prototype

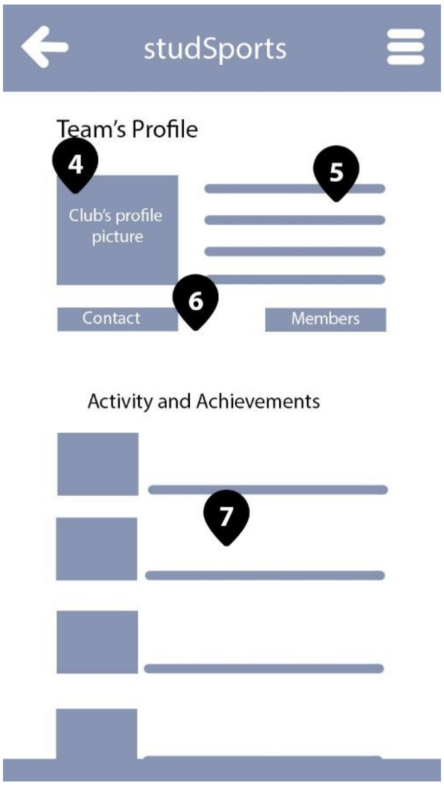
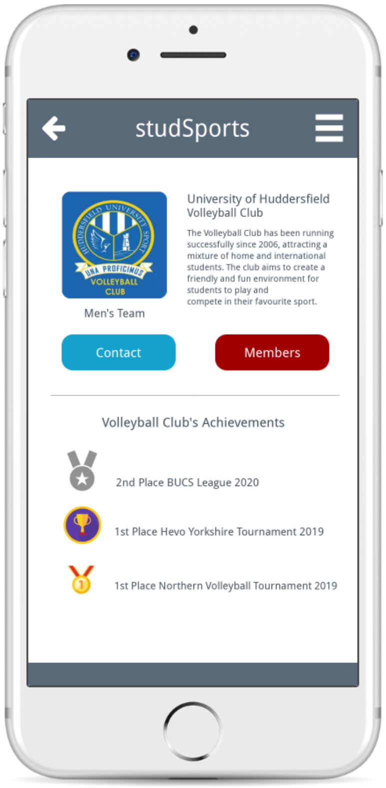


Figure 3.5 Club's Profile Wireframe



University of Huddersfield Volleyball Club

The Volleyball Club has been running successfully since 2006, attracting a mixture of home and international students. The club aims to create a friendly and fun environment for students to play and compete in their favourite sport.

Volleyball Club's Achievements

2nd Place BUCS League 2020

1st Place Hevo Yorkshire Tournament 2019

1st Place Northern Volleyball Tournament 2019

Figure 3.6 Club's Profile Prototype

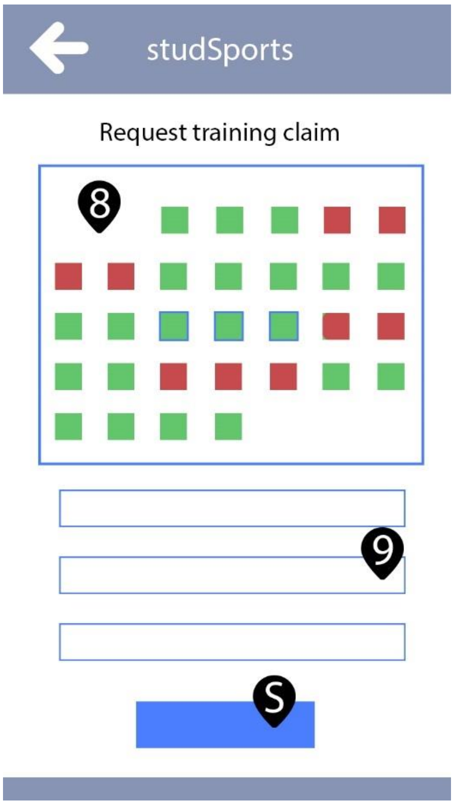
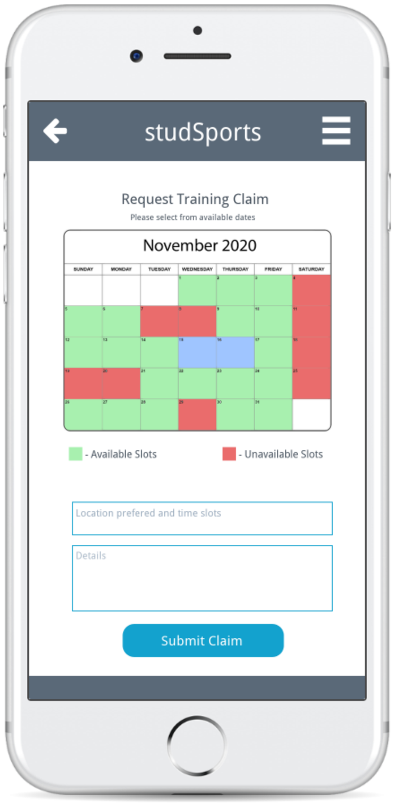


Figure 3.7 Request Page Wireframe



Request Training Claim

Please select from available dates

November 2020

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

🟩 - Available Slots 🟥 - Unavailable Slots

Location prefered and time slots

Details

Figure 3.8 Request Page Prototype

The prototyped task flow had the following user requirements:

- The home page should display the main application services (browse events, sport clubs, tournaments, summer camps, marketplace, advance search)
- The application should display the list of sport clubs registered within the application
- System should display the information about club's activity and achievements
- Users should be able to contact the sport club
- Users should be able to request a common training session

When prototyping the application, I have tried to implement every user requirement, because this decides rather the application is going to be used or not. User requirements represent the audience for whom I am developing the application. Users are the most important part in every development process.

3.3 User Testing Plan

My user testing phase was based on both qualitative and quantitative research, as I have developed a high-fidelity app prototype and I wanted the users to participate and directly observe the prototypes in a natural way. I needed to investigate in deep the data collection and make very detailed specification for my product development. I used both usability and walkthrough testing methods to figure out the problems with the design and receive feedback. Using my laptop and MarvelApp, I asked users to complete specific tasks and go through the application, observing at the same time where do they encounter problems and then check and discuss different solutions to problems and different suggested alternatives. The main aim of this evaluation was to ensure any potential issues are highlighted and fixed in the final product prototype.

I've selected audience from different backgrounds and of different age, in this way the youngest selected user was my teammate, who is 19 years old and studies law, and oldest user was my volleyball coach, who is 36 years old. The rest 4 users were students doing different courses from different sport clubs. The diversity in users' interests is a huge advantage, as it provides a better feedback of the app viewed from different angles.

While doing the user testing, I was trying to listen and ask, rather than answer questions. At the beginning I would always open the prototype in marvelapp and ask the subject to perform a task(e.g. create a request), and observe and analyse attentively their first moves, without answering any questions. In my opinion this first use had the higher value, as it provided an insightful feedback on my navigation system. The second step was asking users to do a free "fly" through the whole application and share their thoughts, the given feedback helped with improving the usability. Lastly, I showed the list of user requirements to the subjects and ask them to evaluate the application once again, focusing their attention on the user requirements satisfaction. I am intending to implement all of my key findings and make a better prototype according to the results from user testing phase and then repeat those three steps with the updated prototype. In this way I can do a better analysis over my results and see how close I am to the final product.

Chapter 4

Development

4.1 Back-End & PHP Frameworks

The back end refers to parts of a computer application or a program's code that allow it to operate and that cannot be accessed by a user. Most data and operating syntax are stored and accessed in the back end of a computer system. Typically, the code is comprised of one or more programming languages. The back end is also called the data access layer of software or hardware and includes any functionality that needs to be accessed and navigated to by digital means [13]. PHP (Hypertext Pre-processor) refers to back-end programming language.

Initially, I was planning on using plain PHP for developing this application, however I changed my mind after researching on PHP frameworks benefits. There are several reasons why I choose to give PHP frameworks a chance. First, PHP frameworks increase the productivity of programming because writing a piece of code that usually takes hours and takes hundreds of lines of code can be done in minutes using built-in functions of the framework [8]. It also provides a framework for the application, usually based on the use of the Model View Controller (MVC) architectural pattern. According to author [7], this ensures a separation between: the data model that is in charge of access and requests to the database, the "view", which is a description of the presentation of the data (ex. HTML), and the controller, which implements the logic by retrieving the model data and sending it to view port. I have considered three of the most popular PHP frameworks used in web industry: Symfony, Yii and Laravel.

a. Symfony

Symfony is a set of reusable PHP components, enabling the developer to create scalable, high-performance applications. With 30 components from which to choose, the developer has the complete freedom to experiment and work in a RAD environment. Symfony APIs also enable easy integration with third-party applications, and it can be used with popular front-end frameworks, such as AngularJS [15].

Many popular projects, including Drupal and phpBB, also use a Symfony framework. In fact, Laravel, the most popular PHP framework, is build off of Symfony.

b. Yii

Created by Qiang Xue in 2008, Yii is a secure, fast, high-performance application/web-development framework. Yii utilizes the Composer dependency manager for PHP for handling different dependencies and installations (more on it later). Yii also is the fastest PHP framework, thanks to the lazy loading technique [15].

Another great feature of Yii is jQuery integration. The integration enables front-end developers to embrace the framework quickly, and it uses scaffolding to generate code. Similar to Symfony, Yii also utilizes components to enable rapid application development.

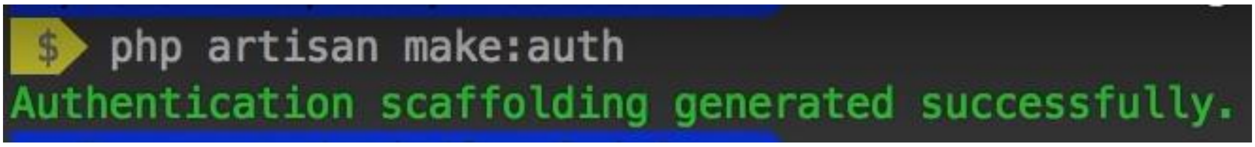
c. Laravel

Laravel is a free, open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) paradigm. Laravel PHP framework was developed with the aim to make a few development tasks easier for developers such as authentication, sessions, caching, and routing.

I have chosen Laravel over the first two frameworks, because it suited better my middle-size type of project, as well as provided the following advantages, which I used in development:

1. *Authentication and Authorization Systems*

One of the most important features in modern web applications is the authentication. Laravel provide a fast and easy way to implement authorization functionality by just writing the following commands in php artisan command line:

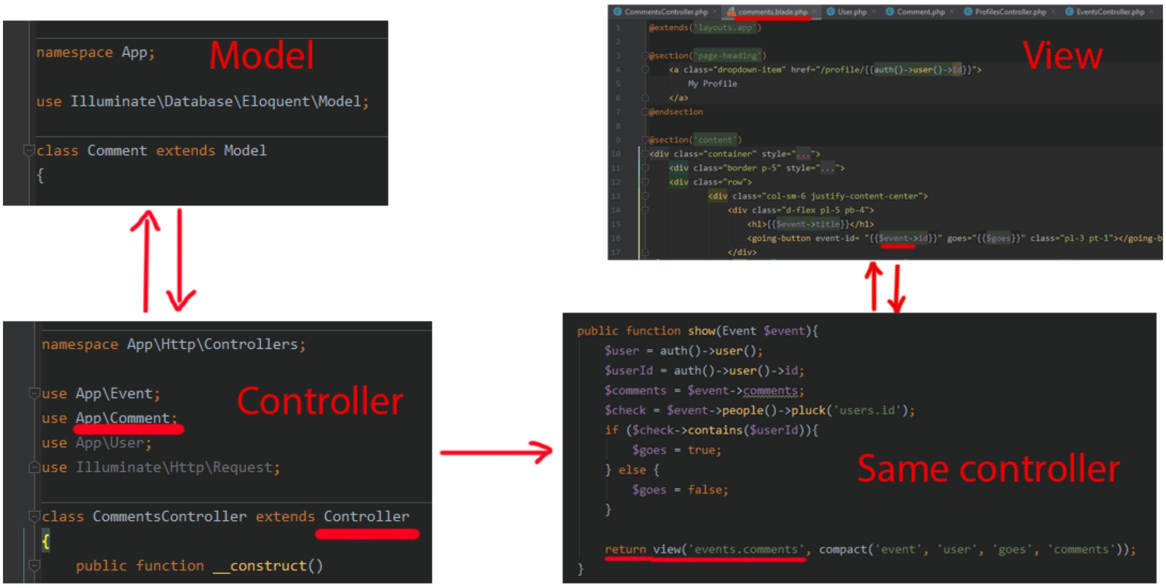


```
$ php artisan make:auth
Authentication scaffolding generated successfully.
```

Figure 4.1 Php artisan command line

After the command is being executed, Laravel automatically creates authorization model, view, and controller. The user table is also created in the database and can be customized to remove or add more fields.

2. *MVC Architecture Support*
   Laravel follows a Model-View-Controller architecture, which enhances the performance, ensures clarity, and allows for better documentation.



```
namespace App;

use Illuminate\Database\Eloquent\Model;

class Comment extends Model
{
```

```
@extends('layouts.app')

@section('page-heading')
    <a class="dropdown-item" href="/profile/{{auth()->user()->id}}">
        My Profile
    </a>
@endsection

@section('content')
<div class="container" style="...">
    <div class="border p-5" style="...">
    <div class="row">
            <div class="col-sm-6 justify-content-center">
                <div class="d-flex pl-5 pb-4">
                    <h1>{{$event->title}}</h1>
                    <going-button event-id= "{{$event->id}}" goes="{{$goes}}" class="pl-3 pt-1"></going-b
                </div>
```

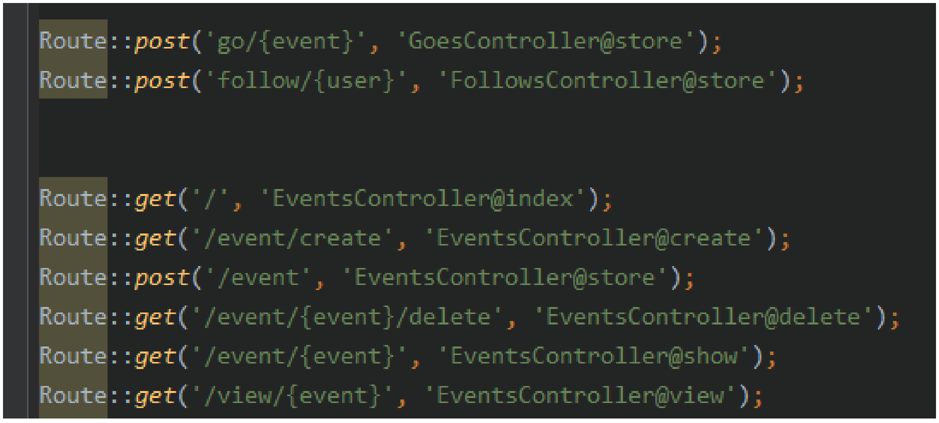
```
namespace App\Http\Controllers;

use App\Event;
use App\Comment;
use App\User;
use Illuminate\Http\Request;

class CommentsController extends Controller
{
    public function __construct()
```

```
public function show(Event $event){
    $user = auth()->user();
    $userId = auth()->user()->id;
    $comments = $event->comments;
    $check = $event->people()->pluck('users.id');
    if ($check->contains($userId)){
        $goes = true;
    } else {
        $goes = false;
    }

    return view('events.comments', compact('event', 'user', 'goes', 'comments'));
}
```

Figure 4.2 Laravel MVC pattern demonstration

Figure 4.2 demonstrates the use of MVC pattern in building the application. The model file creates a class which connects to database, this class is usually bound to a specific table in a database. In my case, Comment class has its table in the database called comments. The CommentsController class extends model Controller class so that it can pass the data from database to the view file (comments.blade.php). View file represents front-end part of the application, users can interact only with “View”. For security reasons Model can communicate only with the Controller, this is made to protect the database from hacker attacks (ex. SQL injections).

3. *HTTP Routing*



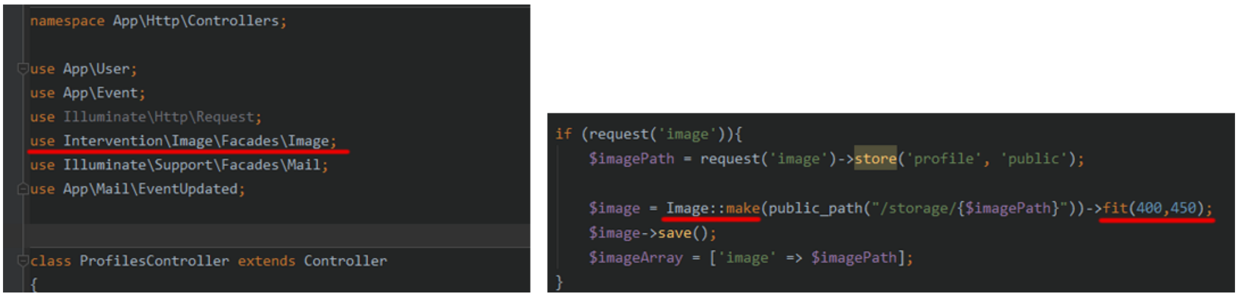
```php
Route::post('go/{event}', 'GoesController@store');
Route::post('follow/{user}', 'FollowsController@store');


Route::get('/', 'EventsController@index');
Route::get('/event/create', 'EventsController@create');
Route::post('/event', 'EventsController@store');
Route::get('/event/{event}/delete', 'EventsController@delete');
Route::get('/event/{event}', 'EventsController@show');
Route::get('/view/{event}', 'EventsController@view');
```

Figure 4.3 Http routes code snippet

Laravel provides a feature called HTTP routing, which is used to control the interactions between users and web application. Every route represents an action and leads to a response to users' action. Routes has also a security role, while using POST or PATCH methods to get the form data, a CSRF token is being passed through making it easy to protect the application from cross-site request forgeries. Cross-site request forgeries are a type of malicious exploit whereby unauthorized commands are performed on behalf of the authenticated user. Laravel automatically generates a CSRF "token" for each active user session managed by the application. This token is used to verify that the authenticated user is the one actually making the requests to the application.[4]

4. *Intervention Image library*

Intervention Image is a PHP image handling and manipulation library providing an easier and expressive way to create, edit, and compose images [6]. The web application is going to have customizable profile pics, thumbnails, and other features which dwell with images. That is why, is important to have a powerful library that is going to rescale the images to keep the overall symmetry in design, saving as much image quality as possible.



```php
namespace App\Http\Controllers;


use App\User;
use App\Event;
use Illuminate\Http\Request;
use Intervention\Image\Facades\Image;
use Illuminate\Support\Facades\Mail;
use App\Mail\EventUpdated;


class ProfilesController extends Controller
{
```

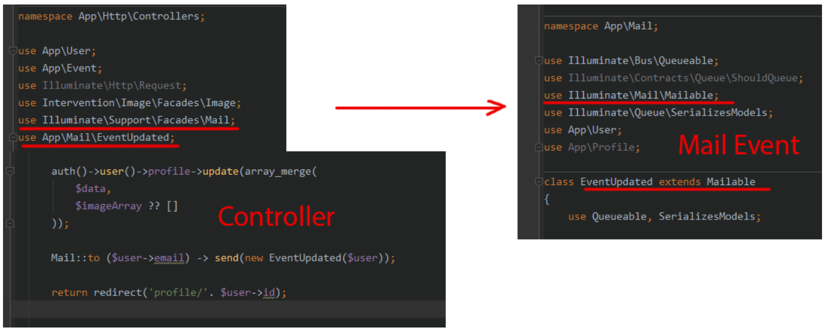
```php
if (request('image')){
    $imagePath = request('image')->store('profile', 'public');

    $image = Image::make(public_path("/storage/{$imagePath}"))->fit(400,450);
    $image->save();
    $imageArray = ['image' => $imagePath];
}
```

Figure 4.4 Intervention Image library use (code snippet)

Intervention Image library can be installed to Laravel using php artisan install command, after the plugin is installed, it is necessary to include it in namespaces of the specific file to start using it. The *figure 4.4* shows how I added the route to Intervention library in the ProfileController file. The library can now be used to rescale the images uploaded by the users, by using the simple method fit(), indicating the needed amount of pixels in the brackets. The library would not only scale the image, but also cut it if the image does not fit the dimensions.

5. *Emails and Notifications*

Every modern application should have a good mail system, that sends notifications about different events directly to user’s email address. Laravel documentation [4] says: Laravel provides a clean, simple API over the popular SwiftMailer library. Laravel also provides drivers for SMTP, Mailgun, Mandrill, SparkPost, Amazon SES, PHP's “mail” function, and “sendmail”, allowing an application to quickly get started sending mail through a local or cloud-based service. In addition to support for sending email, Laravel provides support for sending notifications across a variety of delivery channels, including SMS (via Nexmo) and Slack.



```
namespace App\Http\Controllers;

use App\User;
use App\Event;
use Illuminate\Http\Request;
use Intervention\Image\Facades\Image;
use Illuminate\Support\Facades\Mail;
use App\Mail\EventUpdated;

    auth()->user()->profile->update(array_merge(
        $data,
        $imageArray ?? []
    ));

    Mail::to ($user->email) -> send(new EventUpdated($user));

    return redirect('profile/'. $user->id);
```

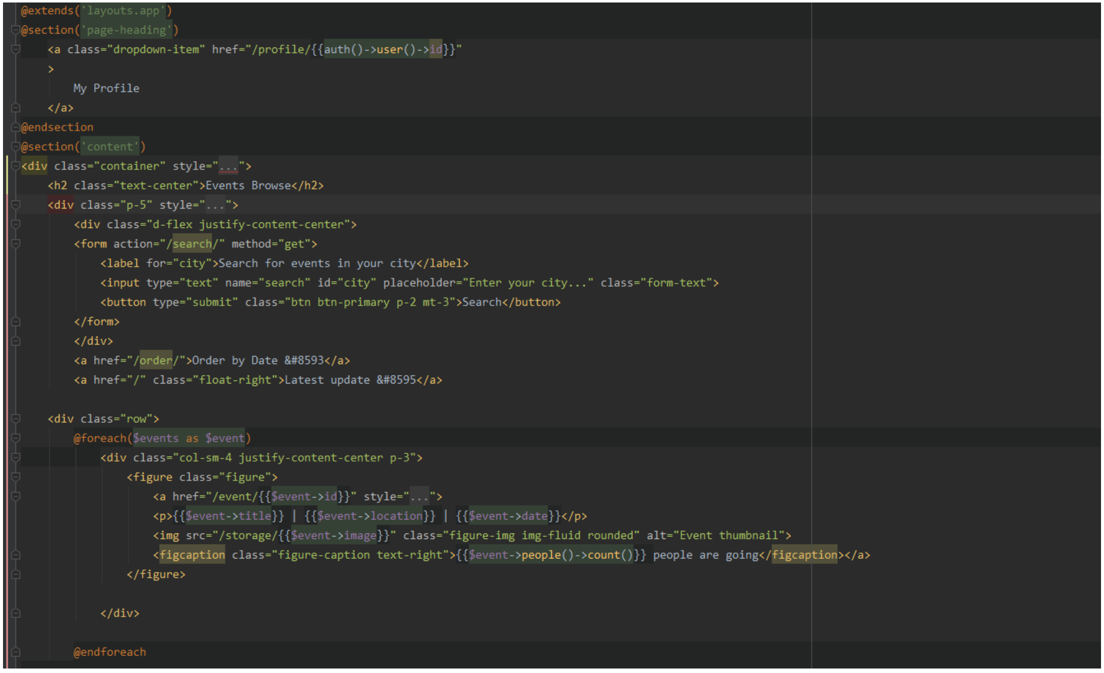
```
namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;
use App\User;
use App\Profile;

class EventUpdated extends Mailable
{
    use Queueable, SerializesModels;
```

Figure 4.5 Mailable Laravel feature (code snippet)

Laravel ships with a feature called Mailable. Mailable accept markdown, which allows to do text formatting without doing any HTML.

6. *Blade Templating*

Blade Templating engine speeds up the process of front-end development by introducing pre-defined templates which are light-weighted, intuitive, and dynamic that allows developers to write code easily and quickly.



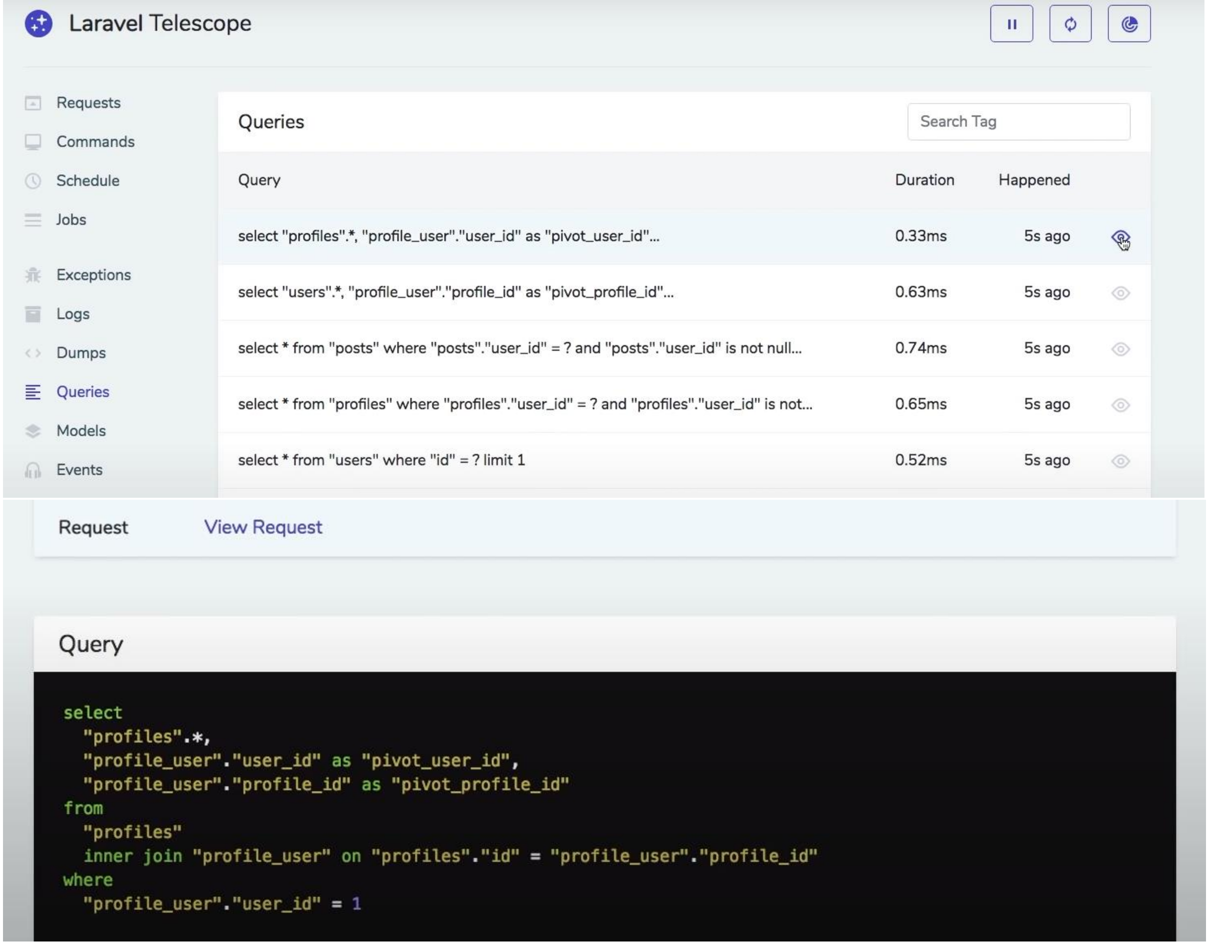
```
@extends('layouts.app')
@section('page-heading')
    <a class="dropdown-item" href="/profile/{{auth()->user()->id}}"
    >
        My Profile
    </a>
@endsection
@section('content')
<div class="container" style="...">
    <h2 class="text-center">Events Browse</h2>
    <div class="p-5" style="...">
        <div class="d-flex justify-content-center">
        <form action="/search/" method="get">
            <label for="city">Search for events in your city</label>
            <input type="text" name="search" id="city" placeholder="Enter your city..." class="form-text">
            <button type="submit" class="btn btn-primary p-2 mt-3">Search</button>
        </form>
        </div>
        <a href="/order/">Order by Date &#8593</a>
        <a href="/" class="float-right">Latest update &#8595</a>

    <div class="row">
        @foreach($events as $event)
            <div class="col-sm-4 justify-content-center p-3">
                <figure class="figure">
                    <a href="/event/{{$event->id}}" style="...">
                    <p>{{$event->title}} | {{$event->location}} | {{$event->date}}</p>
                    <img src="/storage/{{$event->image}}" class="figure-img img-fluid rounded" alt="Event thumbnail">
                    <figcaption class="figure-caption text-right">{{$event->people()->count()}} people are going</figcaption></a>
                </figure>

            </div>

        @endforeach
```

Figure 4.5 Blade Templating Demonstration (code snippet)

I used Blade Templating engine to write most of my view pages layouts, the basic blade commands are written using ‘at’(@) at the beginning, a custom name is written in the brackets, which works as an ID. The ID is used later in the main layout, to describe the position of the code inside the page. What is more, blade provide the feature of loops, which can loop through the database data, automatising the process of displaying huge amount of data.

7. Telescope debug assistant

Telescope is a debug assistant for the Laravel framework. It is a powerful assistant that tracks every request coming into application, such as, database queries, mails and notifications, log entries, cache operations, variable dumps etc. Telescope helps a lot when developing Laravel applications, as it provides a good insight on everything that happens inside the application, making it easy for the developer to spot and fix the issues.



Queries

Query	Duration	Happened
select "profiles".*, "profile_user"."user_id" as "pivot_user_id"...	0.33ms	5s ago
select "users".*, "profile_user"."profile_id" as "pivot_profile_id"...	0.63ms	5s ago
select * from "posts" where "posts"."user_id" = ? and "posts"."user_id" is not null...	0.74ms	5s ago
select * from "profiles" where "profiles"."user_id" = ? and "profiles"."user_id" is not...	0.65ms	5s ago
select * from "users" where "id" = ? limit 1	0.52ms	5s ago

Request View Request

Query

```
select
  "profiles".*,
  "profile_user"."user_id" as "pivot_user_id",
  "profile_user"."profile_id" as "pivot_profile_id"
from
  "profiles"
  inner join "profile_user" on "profiles"."id" = "profile_user"."profile_id"
where
  "profile_user"."user_id" = 1
```

Figure 4.6 Telescope debug assistant

I have used Telescope to handle some errors with database, as the studEvents database is quite complex one, I had to track the requests and responses to understand in which moment the error occurred. By clicking on a query, Telescope will show the full query, as well as an option to view the request.

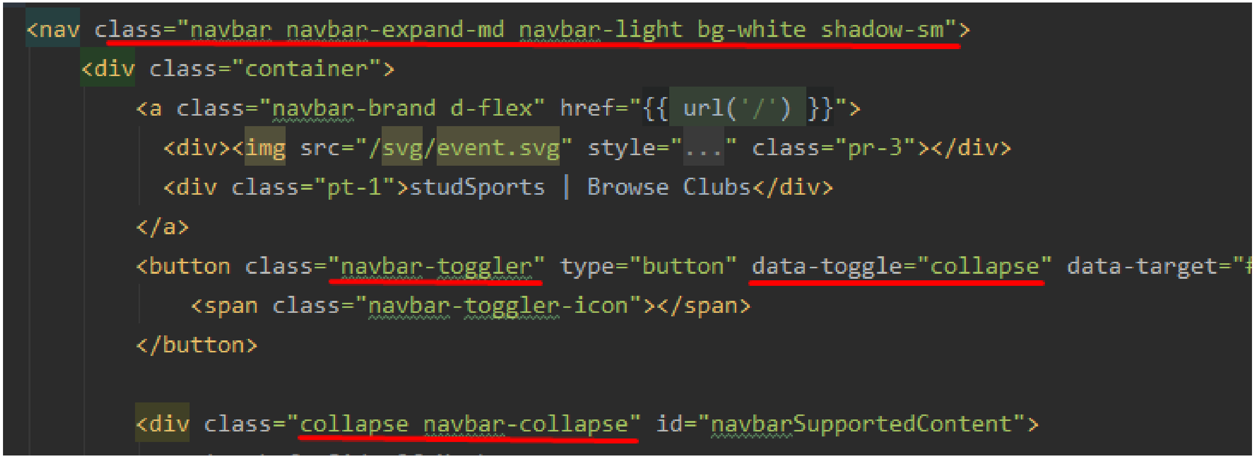
4.2 Front-End Tools

Front-End refers to the side of the web application that can be seen by the user, it is designed using languages like HTML, CSS, and JavaScript. Modern enterprises are using different front-end tools and techniques to speed-up the prosses of development, and make the code more flexible and maintainable, as well as boost the performance of the web application. Some of these tools and techniques include CSS frameworks, CSS pre-processors, Scalable images.

CSS frameworks contains predefined bits of code, that can be used to build website faster and easier, as well as making the layout clearer. There is a deluge of CSS front-end frameworks available nowadays. But the number of good ones can be narrowed down to just a few. The most popular CSS framework in 2020 is Bootstrap, which hold more than 130,000 stars on GitHub [3]. I have decided on using Bootstrap for my project, because I am familiar with it and it offers a range of benefits that perfectly suits my application that must be mobile responsive.

a. Bootstrap

Bootstrap enables front-end developers to quickly build fully responsive websites. The biggest advantages of Bootstrap are responsive grid system, responsive images, customizability. Bootstrap makes creating grid for the application very easy, as it has its own grid system predefined. What is more, the grid created with Bootstrap is responsive, which means the content is going to change accordingly to the user device's screen size. To make the images responsive, it is enough to add a simple class img-responsive to the images tag inside the HTML file.



```
<nav class="navbar navbar-expand-md navbar-light bg-white shadow-sm">
    <div class="container">
        <a class="navbar-brand d-flex" href="{{ url('/') }}">
          <div><img src="/svg/event.svg" style="..." class="pr-3"></div>
          <div class="pt-1">studSports | Browse Clubs</div>
        </a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">
```

Figure 4.7 Bootstrap Navigation Bar (code snippet)

Figure 4.7 represents a code snippet of studSports navigation bar, by inserting a number of pre-defined Bootstrap class names into nav tag, I have saved a lot of time and effort. The navbar-toggler class and collapse class provide the "hamburger" menu feature, by typing on the three lines icon, the navigation will expand.

Bootstrap has a giant collection of predefined classes, this makes the CSS file huge, which can negatively affect the performance of the application. But Bootstrap allows developers to solve this drawback by customizing which functionality is going to be included in application. StudSports uses a minimalistic design, it does not require all of the Bootstrap features, that is why by customizing and cutting off the useless features, the Bootstrap CSS file will become lighter, thus increasing the loading speed and overall performance of the application.

b. SASS

CSS pre-processors are scripting languages that extend the default capabilities of CSS. They enable front-end developers to use logic in CSS code, such as variables, nesting, inheritance, mixins, functions, and mathematical operations. [9]

According to State of CSS 2019, the mainstream CSS pre-processors are Sass, Less, PostCSS and Stylus.

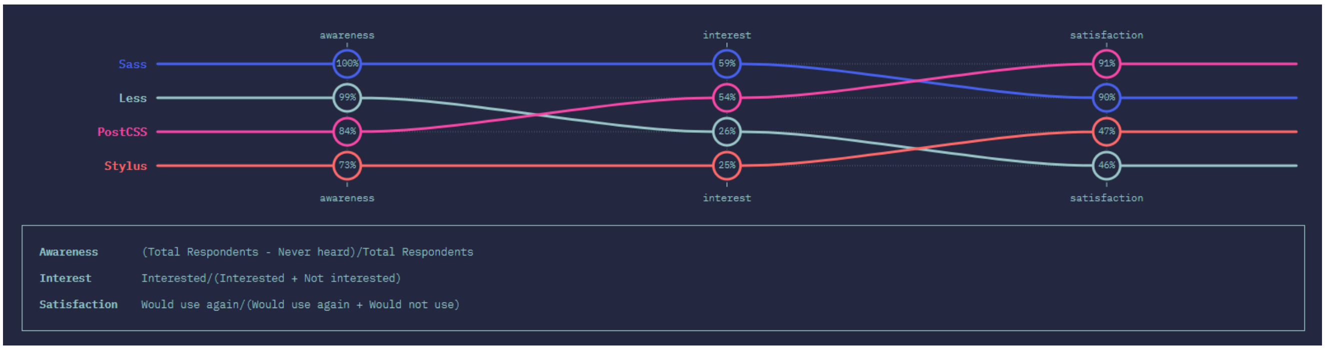
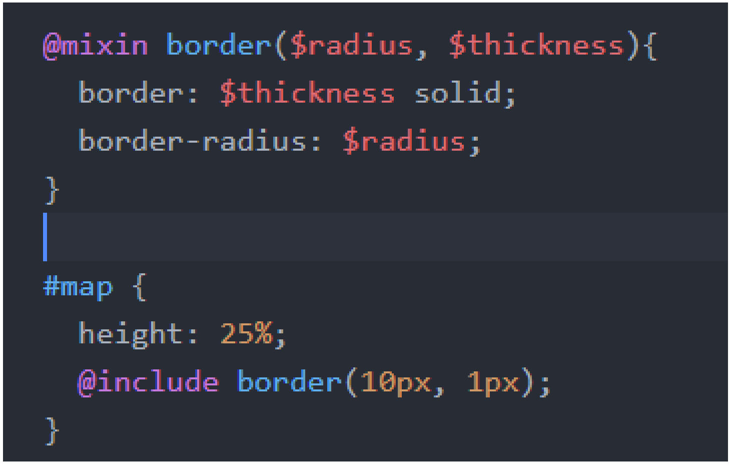


Figure 4.8 CSS pre-processors statistics

The mainstream in 2020 is SASS (Syntactically Awesome Stylesheets, that is why, to keep up with the modern world, I chose to use it for my project.

SASS has specific style standards which are meant make the writing of the document even easier, such as the freedom to omit braces. It lets you use variables, mathematical operations, mixins, loops, functions, imports, and other interesting functionalities that make writing CSS much more powerful and easy to maintain [11]. SASS extends standard CSS abilities by introducing the benefits of a basic programming language. Later SASS will compile the code and generate CSS output a browser can understand. SASS eliminates the need of repeating a piece of code every time by introducing variables. For example, if the main colours of application are assigned to variables, it is easy to change the whole colour scheme of the app by editing variable values. Variables and other SASS features, such as, mixins, loops and functions make the writing of CSS code less time consuming, and much easier to maintain.



```
@mixin border($radius, $thickness){
  border: $thickness solid;
  border-radius: $radius;
}

#map {
  height: 25%;
  @include border(10px, 1px);
}
```

Figure 4.8 SASS pre-processor code snippet

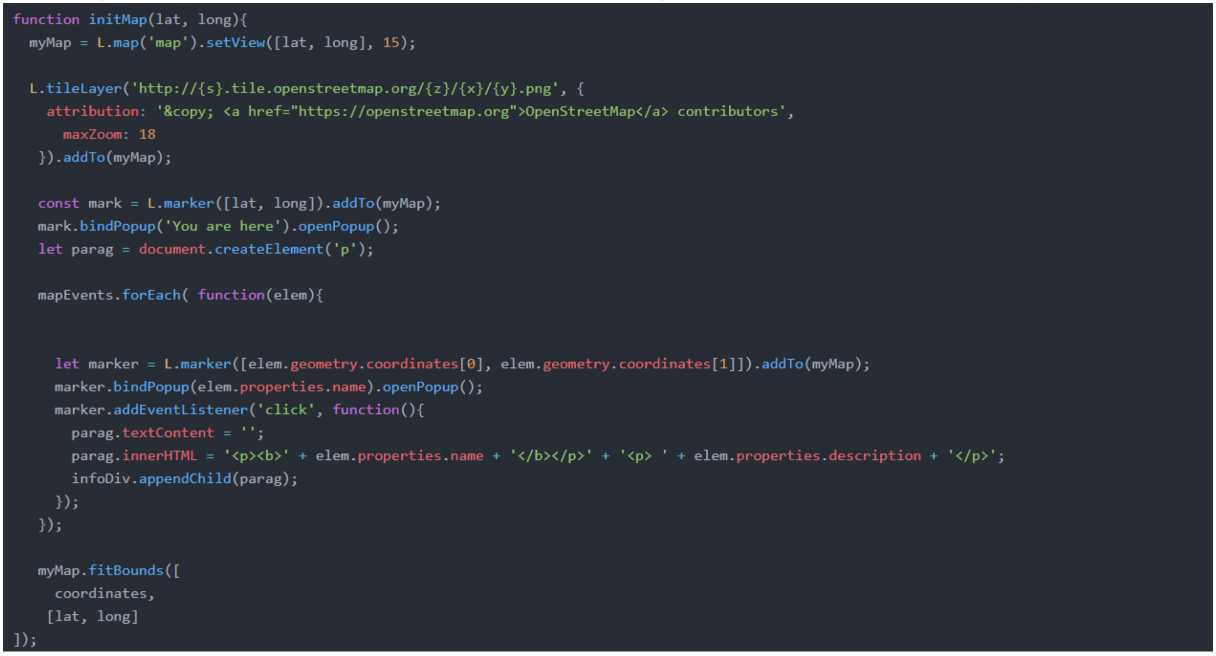
Figure 4.8 shows an example, where I wrote a mixin for border property, later on I used this mixin to give the map id a border with 10px thickness and 1px radius, by inserting the values in the mixin parameters.

c. SVG Images

SVG (Scalable Vector Images) are defined in text XML files, rather than using pixel grid, SVG uses shapes, numbers, and coordinates to render graphics in the browser, which makes it perfect in terms of scalability, responsiveness, and performance[12]. Even though SVG image is much lighter than JPG, JPEG, PNG, or other images formats, it does not lose quality when it is rescaled. Also, the use of SVG icons will greatly increase the overall loading speed of the application. StudSports uses a minimalistic design, that is why, I have tried to make a good use of SVG, replacing the icons and some of the thumbnails.

d. Leaflet JavaScript Library

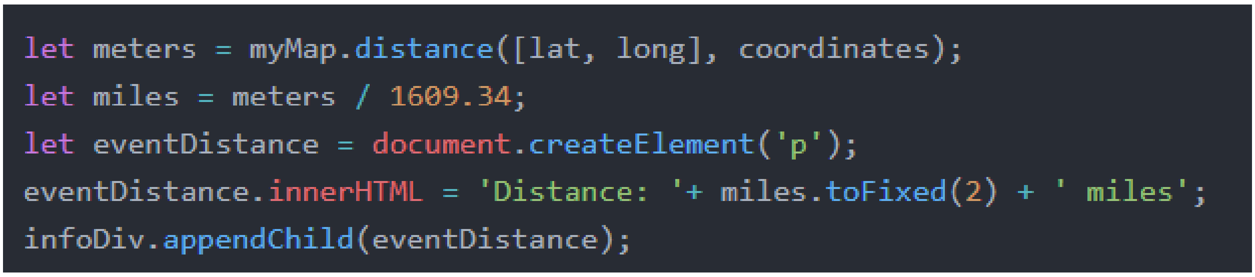
StudSports application is going to provide directions to different locations, such as, events location, universities locations, sport courts location. Leaflet open-source JavaScript library allows us to easily implement mobile-friendly interactive maps to the application. I have chosen it, because the JS file is very light, but it has all the important mapping features, which make this library suitable to use for studSports application development. This library makes implementing the mapping features to the application very easy. The map can also be restyled to make it look unique. One of the most important things about this library, is the performance it shows while working on mobile devices. The scroll and zooming feels as smooth as native apps.



```
function initMap(lat, long){
  myMap = L.map('map').setView([lat, long], 15);

  L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    attribution: '&copy; <a href="https://openstreetmap.org">OpenStreetMap</a> contributors',
      maxZoom: 18
   }).addTo(myMap);

   const mark = L.marker([lat, long]).addTo(myMap);
   mark.bindPopup('You are here').openPopup();
   let parag = document.createElement('p');

   mapEvents.forEach( function(elem){


     let marker = L.marker([elem.geometry.coordinates[0], elem.geometry.coordinates[1]]).addTo(myMap);
     marker.bindPopup(elem.properties.name).openPopup();
     marker.addEventListener('click', function(){
       parag.textContent = '';
       parag.innerHTML = '<p><b>' + elem.properties.name + '</b></p>' + '<p> ' + elem.properties.description + '</p>';
       infoDiv.appendChild(parag);
     });
   });

   myMap.fitBounds([
     coordinates,
    [lat, long]
]);
```

Figure 4.9 JavaScript code snippet

Picture 4.9 shows how easy it is to initiate the map, knowing user position. The library also simplifies the process of adding markers to the map, by simply selecting the marker and writing the addTo(map) method, in which map is the initiated map. Leaflet also provides fitBounds() method, which scale the map, so that it shows every marker position without zooming in or out.



```
let meters = myMap.distance([lat, long], coordinates);
let miles = meters / 1609.34;
let eventDistance = document.createElement('p');
eventDistance.innerHTML = 'Distance: '+ miles.toFixed(2) + ' miles';
infoDiv.appendChild(eventDistance);
```

Figure 4.10 JavaScript code snippet

This code snippet shows how easy it was to calculate and draw distance to some points using Leaflet.

Chapter 5

Evaluation and Conclusion

5.1 UR Evaluation

In this table I evaluate rather the User Requirements were fulfilled or not.

User Requirement	Status
Users should be able to find contacts of sport clubs commitment members	Fulfilled
Users should be able to request a common training session with other university team	Fulfilled
Users should be able to organize friendly games or matches with other universities using the application	Working
Users should be able to share activity and achievement of their club	Fulfilled
Users should be able to see the full list of sport clubs registered within the application	Fulfilled
Users should be able to browse events	Fulfilled
Users should be able to advertise their events	Fulfilled
Users should be able to do filter the search results	Fulfilled
Users should be able to create listing and sell sport items	Fulfilled
Users should be able to view existing listings and contact the seller	Fulfilled
Users should be able to create auction on their listings	Working
Users should be able to place bids on listings	Working
Users should be able to join a tournament	Fulfilled
Users should be able to create a tournament	Working
Users should be able to track the tournament progress	Working
Users should be able to edit their profile	Fulfilled
Users should be able to update their profile picture	Fulfilled
Users should be able to access another user profile	Fulfilled
System should send welcome notifications to new users	Fulfilled
System should notify users about updates on their club	Working
System should notify users about updates on their requests	Working
Users should be able to register in the application	Fulfilled
Users should be able to authenticate to application	Fulfilled
Users should be able to advertise sport summer camps and tours	Working

Table 5.1 User Requirements Evaluation

5.2 Project Evaluation

In this project I have used most of my Information Technologies knowledge and skills to develop a web-based application for university students. I have encountered a series of difficulties regarding the database design, which I overcome by studying the university materials from my 1st and 2nd year. Some difficulties using Laravel have appeared, these were regarding the many-to-many relationships inside the application classes. As the database was very complex, it was a meticulous work to make the data flow properly inside an MVC architectural pattern. The advice that I would give to the past myself, would be to not try to incorporate too many different services in one application, as it makes the database very complex. But I am glad nobody gave me this advice, because I learned a lot while dealing with complex database tasks, sometimes I was enjoying finding solutions to them. This project helped me choose the future job of Back-End programmer, which I didn't consider before.

5.3 Conclusion

Many sport students around the UK are unaware of the opportunities they lose due to lack of communication with other universities. In this project I came with a change to universities sport system, I have developed an application that is going to facilitate the communication between universities sports clubs. In this way, many students will not only growth in a better sport environment, allowing them to train and compete on daily basis, developing their professional skills, but also profit from social events organized by students into sport where they can socialize with people that have same passion. In a competitive environment, there is a big chance of educating talents, that may later become world famous sport professionals. Socializing and improving communication skills will also give student huge benefits, including a better preparation for future career. I hope that one day the application presented in this project will become available to the whole country, and that it will make students' lives better.

Even though, the development of web-based applications using modern IT technologies is a complex process and requires a lot of research and technical skills, with a good aim and motivation the process becomes enjoyable. This project has changed my attitude upon dwelling with complex tasks, it also improved my technical and academic skills.

5.4 Future Plans

I successfully realised my aim and developed a Laravel application for students interested in sports. Unfortunately, some of the objectives were not achieved due to lack of time. But I promised myself that in the nearest future I will start over this project, making the application available on mobile devices using both android and iOS operating systems. I am going to fulfil every user requirement and open the application to the public. The first users are going to be my friends and teammates from volleyball club. After testing the application on them, I will advertise it to the whole university of Huddersfield, and later on to the whole country.

Appendix

[1] Nielsen’s Usability Heuristics

Visibility of system status: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time
Match between system and the real world: The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order
User control and freedom: Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support ‘undo’ and ‘redo’. Home and back controls should be also present.
Consistency and standards: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. Icons should be recognisable and consistent.
Error prevention: Even better than good error messages is a careful design, which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action
Recognition rather than recall: Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
Flexibility and efficiency of use: Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
Aesthetic and minimalist design: Dialogues should not contain information, which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
Help users recognize, diagnose, and recover from errors: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution
Help and documentation: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large

Table appendix 1.1 - Nielsen’s Usability Heuristics

References

[1] A. Mishra and D. Dubey “A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios” vol 1, pp 1-4, 2013

[2] R. Friis Dam and T. Yu Siang “Personas – A Simple Introduction”, 2020 Available on: https://www.interaction-design.org/

[3] Bootstrap GitHub repository, Available on: https://github.com/twbs/bootstrap

[4] Laravel Documentation, Available on: https://laravel.com/docs/7.x

[6] Intervention Image Documentation, Available on: http://image.intervention.io/

[7] B. Khaoula, “A new model for the selection of web development frameworks: application to PHP frameworks” Vol. 9, No. 1, pp. 695~703, 2019

[8] N. Prokofyeva and V. Boltunova, “Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems,” Procedia Comput. Sci., vol. 104, pp. 51–56, 2017.

[9] A. Monus “Popular CSS Preprocessors With Examples: Sass, Less & Stylus”, 2018 Available on: https://raygun.com/blog/css-preprocessors-examples/

[10] State of CSS Statistics 2019, Available on: https://2019.stateofcss.com/

[11] SASS documentation, Available on: https://sass-lang.com/documentation

[12] I. Gerchev “SVG 101: What is SVG? (and what is it actually good for?)”, 2017 Available on: https://www.sitepoint.com/svg-101-what-is-svg/

[13] Front-End and Bck-End definition, Available on: https://whatis.techtarget.com/definition/front-end

[14] V. Massey, K.J Satao, “Comparing Various SDLC Models and The New Proposed Model On The Basis Of Available Methodology”

[15] Michael J. Garbade “How to choose a PHP framework” 2016 Available on: https://opensource.com/business/16/6/which-php-framework-right-you