UNIVERSITY OF PLYMOUTH

School of Computing, Electronics and Mathematics

PRCO304
Final Stage Computing Project
2019/2020

BSc (Hons) Computer Science

Daniel Thick
10555972

CloudPT

Supervisor: Marius Varga
Second Marker: Thomas Heinzl

ACKNOWLEDGEMENTS

I would like to thank my project supervisor, Marius Varga, for all the help and support he has provided throughout the completion of my final year project.

I would also like to thank my girlfriend, family, and friends for their continued support throughout this project as well as the entirety of my university degree.

Finally, I would like to thank all of the users that participated in the usability testing for providing valuable feedback on the application.

ABSTRACT

This report describes a software design and development project for a progressive web application that can be used by personal trainers to help communicate with their online clients. The application allows for users to be able to instant message, create and assign workouts, record workouts, and track progress.

The report begins by first identifying the problem, who the client for the project was, and the aims and objectives. This was then followed by conducting some background research into existing products on the market and their limitations, as well as market research about the demand for this kind of product. The way the project was managed is then discussed, including any legal, social, and ethical issues that needed to be considered. Initial designs and specification for the architecture and user interface were then spoken about. Lastly, an investigation then took place into what development technologies should be used and why.

The implementation process is then documented and broken down into the separate sprints that were completed. Each sprint has a summary, deliverables, and discussion to give the reader a full understanding as to how the development process took place. This then leads to describing the testing that took place during development, and how it was useful and improved the application.

Finally, the report ends by giving a complete evaluation on how the project went. This includes discussing whether requirements were met, decisions that were made, the developer's performance, and further development. The report also includes appendices that are referenced throughout the report's main body for further information and a user guide.

TABLE OF CONTENTS

Word Count: 9,608

OneDrive Submission Link:
https://liveplymouthac-my.sharepoint.com/:f:/g/personal/daniel_thick_students_plymouth_ac_uk/ElDEnE_NLnFBjUwPRQi7j4wBnUSqKhuVj6CX4TZ9z-y_lQ?e=rg4x5G

GitHub Repository:
https://github.com/danthick/PRCO304

1 INTRODUCTION

1.1 PROBLEM IDENTIFICATION

The fitness industry has become one of the fastest growing industries over the last decade, with it now being worth over £25bn in Europe alone. The sector gained 2.4 million new members throughout 2019 in Europe, bringing the European total amount of people with gym memberships to 64.8 million (Rutgers *et al*., 2020). With lots of new people joining the gym, comes lots of people not knowing what to do in the gym and requiring a personal trainer. This has resulted in a large number of new personal trainers as well, with the UK currently having 57,000 registered, which is expected to be growing at a rate of around 2.3% annually since 2015 (Future Fit Training, 2019).

Many personal trainers are shifting their business model to be able to work online rather than face to face with clients. From 2014 to 2017, health and fitness application usage grew over 330%, with 75% of active users opening a fitness application at least twice a week (Saye, 2018). This is because it can significantly reduce costs to both PTs and their clients, which in turn, makes having a PT much more appealing to many people. Furthermore, it allows for PTs to communicate with their clients much more frequently without using up so much of their time. Online personal training also has many benefits for clients as well; it is very flexible, and programs can be designed to work around what is best for them. It allows for them to have a much wider selection of available personal trainers and can improve motivation.

Many personal trainers who are working with an online client base use social media platforms, such as Instagram (Instagram, 2020) and Facebook (Facebook, 2020). These platforms are not only used for communication with their clients and give them the information they need for their workout programs, but also to help find new clients (Urbanek, 2017). This can and does work for many personal trainers, but it is not the best solution possible and a dedicated application can improve their business as well as save them time.

1.2 THE CLIENT

The issue that this application plans to solve was introduced whilst discussing the shift towards online workout programs with a personal trainer. This particular personal trainer has a predominantly online client base and has issues on keeping track with their client's workouts and progress. My client currently uses Instagram's (Instagram, 2020) direct messaging to contact his clients as well as Google Sheets (Google, 2020) to create workouts which can then be sent to the client. This is not the most suitable solution for them as it is very difficult for them to stay on track with what workouts are assigned to which clients and also how their clients are progressing over time.

Having a real client for this project has proven to be very helpful throughout the whole process. It has allowed for the objectives and aims to be refined specifically to what the

client's needs were and ensured that the deliverables have been met. Working with the client during the implementation process helped maintain the scope of the project and allowed for objectives to be changed if the client desired this.

1.3 Aims and Objectives

1.3.1 Aims

The aim of this project was to complete the development of a progressive web application (web.dev, 2020) that would provide a platform for personal trainers to work with their clients online. This allows for personal trainers and clients to communicate with ease and for workouts to be assigned, logged, and monitored. Developing a progressive web application means that it will be compatible regardless of the user's browser and device, as well as responsive by adapting to all screen sizes. Furthermore, it behaves very similar to native mobile applications and will even be installable and have some functionality offline.

1.3.2 Objectives

- Identify problem and complete market research
- Analyse user stories to gather requirements for the application
- Setup continuous integration and delivery
- Design and implement database schema
- Design and implement API/Middleware
- Design and implement a progressive web application

2 Background

2.1 Existing Products

Before starting the design stage of the application, research was completed on products that are already on the market and are trying to solve a similar issue. A comparison table was used (Appendix 15.6) for this situation as there were multiple products already available, each with very different features. By using a comparison table there is a direct side-by-side comparison of many products which helps determine what features are going to be needed and why. Four similar products were found: My PT Hub, TrainCoach, Trainerize and FitHub. These had very similar core functionalities but overall, they are very different applications.

My PT Hub is the most popular application for personal trainers, with over 85,000 fitness professionals signed up (My PT Hub, 2020). It is also the most polished application with the most features included out of the ones that were compared. My PT Hub allows for personal trainers to create workouts and nutrition plans for clients to follow. However, it is business orientated and includes a marketplace for workouts,

videos and nutrition plans that have already been created. It also allows for personal trainers to be able to customise the application to include their own logo and colour scheme to help them build a brand.

TrainCoach, Trainerize, and Fithub all offered the same basic functionalities of a workout builder and tracker. Other than these features, Fithub is a very limited application and does not have many more. In comparison, Trainerize and True Coach both include more features such as instant messaging, results tracking and nutrition tracking. For more information on the research that was conducted, please see (Appendix 15.6).

Using the comparison table helped with getting a better understanding of what features are needed to create a minimum viable product. Furthermore, it was also used to plan for how a new solution should be different and what would make the current market better.

2.2 Limitations of Current Solutions

After completing research on existing products, some limitations of current market became clear. One of the main limitations was the lack of two-way communication between personal trainers and their clients. In most of the products that were researched, personal trainers are able to make notes and comment on a workout, but a client is not able to do the same after completing a workout. This means that the trainer would not know how their client feels about how their workout went. This application will therefore include the ability for both clients and personal trainers to leave feedback as well as an instant messaging system to help communication between the two parties.

Another limitation that can be seen in existing products is the large monthly charge, which is proven with the market research that was conducted for this report. Having a monthly charge for users pushes them away and creates a situation where they will make do with free solutions, such as social media and spreadsheets. As a result, this application will be free to use for all users.

2.3 Market Research

To conduct some initial market research, a survey was created and posted in a few personal training forums to help get an idea of what other personal trainers were using in terms of software to interact with their clients. The questions and results of the survey can be seen in (Appendix 15.7). The results help achieve a better understanding as to what software personal trainers are using and why. From the results it can be seen that the majority of personal trainers, just like my client, use social media to contact their clients, and spreadsheets to give clients their workouts. It can also be seen that personal trainers find it difficult to track a client's progress.

More personal trainers than originally thought had never considered using a dedicated application to help communicate with their clients. The trainers that considered it but do

not use a dedicated application is mainly because of the monthly cost that a lot of products charge. In addition to this, the monthly charge also increases as the personal trainer's client base grows. Completing this survey shows that a lot of personal trainers would consider using a specific application to interact with their clients and build workouts.

3 METHOD OF APPROACH

3.1 INITIAL PLANNING

At the start of the project, an initial plan was created, to help clarify what the aim of the project was and how it was going to be achieved. A project plan was developed to identify specific goals that needed to be achieved and by what date. This was also used to ensure that the project was feasible with time and technology restraints. Furthermore, the initial plan helped find what new skills and technologies would have to be learnt for the implementation stage. A risk assessment was also performed to help prepare for potential issues that may have occurred during the project. This was important to complete because it shows factors that may have an effect on the schedule of the project. Coming up with a strategy for each potential risk before they happen will help reduce the effect they have. For more information about the initial planning, please see (Appendix 15.11).

3.2 AGILE METHODOLOGY

An agile approach was used for both the project management and the development process of this project. Using an agile approach means valuing:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

(Beck *et al*., 2001)

Using an agile approach really helps with meeting requirements and improving the client's satisfaction with the product as it allows for the product to be improved with each iteration using the clients feedback. Changes can be made quickly as feedback is received often due to the short iterations; this prevents the project from being delayed if requirements change. At the end of each iteration, a review takes place to look back at what went well and what might need improving. This can help understand what changes need to be made to future iterations if needed which helps anticipate changes to the project schedule. Having a review at the end of each iteration improves the visibility of the entire project as it allows for time to reflect on what has and has not been completed so far. Furthermore, agile allows for a fast and flexible approach which helps increase

productivity and prioritise requirements easily. Therefore, ensuring that a minimum viable product is delivered on schedule.

3.3 SCRUM

As well as an agile methodology, a scrum framework was also used. Scrum supports development and project management by encouraging teams to learn from their experiences and continuously reflect on what went well and what did not to help improve (Atlassian, 2018). Figure 1 shows how a scrum approach can help turn a product backlog into a finished product.

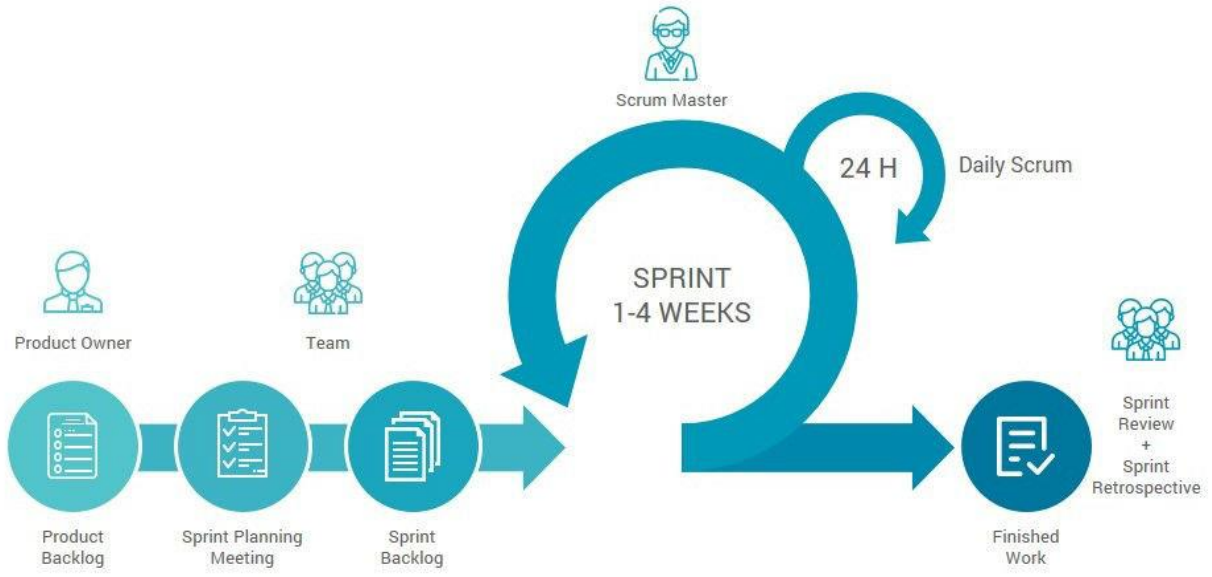


Figure 1. Diagram showing a scrum plan. (Pavel Kukhnavets, 2019).

Using the scrum framework for this project has helped keep track of deliverables and ensure that all requirements were met on time. To achieve this, all the product backlog items were assigned to specific sprints. Furthermore, having sprint reviews at the end of each sprint helps to continuously improve project management and development techniques.

Scrum allows for the client to use the product before it is finished, meaning that feedback can be received and acted upon quickly, keeping client satisfaction high. This is possible because at the end of each sprint there should be a working product that is produced with more features implemented compared to the previous sprint. Having the client test the product throughout the entire process, results in a finished product that has less bugs.

3.4 Test-Driven Development

Test-driven development (TDD) is another agile methodology that was used throughout the development process of this project. TDD works by first writing a test for some functionality and then confirming that the test fails. The next step is to write code to pass the test and confirm that the test passes. The final step is to is to refactor, which includes simplifying the code base and ensuring that all other tests still pass. This cycle, which is shown is Figure 2, is repeated for each piece of code that can be isolated by itself.

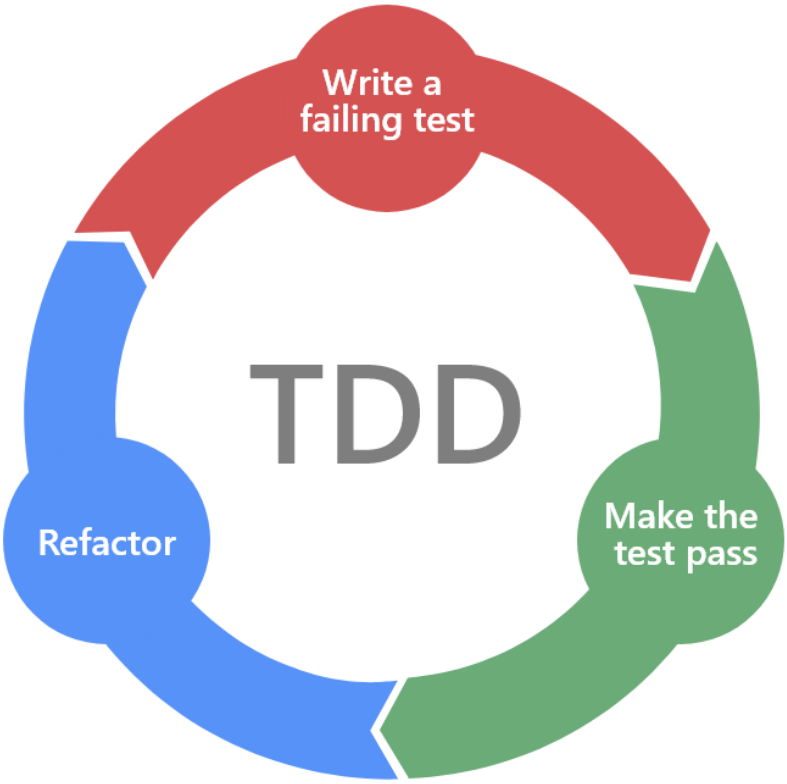


Figure 2. Test Driven Development Cycle. (Marsner Technologies, 2019).

TDD helps to develop useable software as it changes the focus to passing tests and meeting requirements. Using tests significantly reduces bugs during development, but also bugs that do occur can be found and rectified much quicker. Having automated tests can help to identify bugs and other problems that arise due to changes that have been made to different parts of the application. Furthermore, code duplication can be significantly reduced as all code is refactored after it passes the required tests. This also helps to make code more legible and can reduce overall development time by ensuring that the code is of good quality.

4 PROJECT MANAGEMENT

4.1 MEETINGS

Supervisor meetings were held regularly throughout the project to allow for students to discuss any issues and get feedback. For the first month, meetings were held every week and then every other week after that. These meetings were very useful as it gave people a chance to reflect on what had been achieved so far by discussing it with others as well as ensuring that the right path was being followed in terms of project management and development time. As well as getting support from our supervisor during these meetings, it was also a good chance to interact with other students. This allowed students to talk about how their project was progressing and get support from their colleagues. Furthermore, it was a good opportunity to setup testing of project with other students.

Another regular meeting that was help throughout the project were ones with my client. These meetings were held every two weeks via video call and only lasted for approximately 20 minutes each. These meetings would consist of the developer discussing new features that had been implemented and giving the client a chance to test them out. The client would then give any verbal feedback they might have about the new features. In addition, a short discussion would usually take place about the overall direction of the project and if the client had any new requirements.

4.2 VERSION CONTROL

Throughout the project, GitHub (GitHub, 2020) was used to maintain version control for all changes that were made to the code. Version control is very important during development as it allows for developers to track changes and roll back to previous versions if necessary. Having every version of an application means that it is possible to compare different versions of code and see where differences are. In addition to this, it gives the developer confidence to try and makes changes that could cause issues as rolling back changes is easy.

Using GitHub means that the latest version of the code is always stored on their servers. This makes it very easy to work across multiple machines as the current version of code can be pulled and then worked on using any machine. In terms of project management, version control helps to see how much development work is being completed and dates in which features were committed to the application. Although not applying to this project, using version control drastically improves collaboration between developers by creating separate branches and working on the same application at once. For more information about how GitHub and version control was used, see (Appendix 15.12).

4.3 Continuous Integration and Delivery

Continuous integration and continuous delivery (CI/CD) are agile methodologies that are used to speed up deployment. Continuous integration is used to automate the build and testing stage of an application every time a code change is committed to the repository. Continuous delivery will then automate deploying the application to a server, assuming that all the tests pass.

Having a CI/CD strategy in place means that code is built and tested much more frequently and with smaller code changes. This results in issues being much more isolated and can mean that they are fixed quicker than usual. It also helps to have a frequent release rate by always pushing all successful builds to deployment, which keeps customer satisfaction high and maintains an agile methodology. Furthermore, it is straightforward to maintain once up and running and will only alert the developer if a build fails.

For this project, Travis CI (Travis CI, 2020) was used for the build and test server. It connects to the project GitHub repository and runs the build and tests whenever a git commit is pushed to the repository. If the build is successful, then Travis CI will then push the build to Heroku (Heroku, 2020), which is where the application is being hosted. For more information about how CI/CD was implemented in this project, please see (Appendix 15.13).

4.4 Trello

To help with time management, Trello (Trello, 2020) was used to create a project board and keep track of all deliverables. Trello has a very simple set up process which means that within minutes it is ready to be able to input all of the required information. It uses lists which contain cards that can be created with text and images. In addition, it allows for checklists, deadlines, and attachments to be added to each card. This was very useful to be able to check the progress and deadline of each requirement quickly and allowed for deliverables to be met on time as well as nothing being forgotten about.

The project board was set up to have a breakdown of each sprint and which product backlog item would be implemented in which sprint. All the minimum viable product requirements were assigned to a sprint at the beginning of the project, whereas additional functionality was kept in the product backlog list. Furthermore, each requirement was labelled either as backend, design, or feature. This helped organise them and would help other people viewing the board. To see how a Trello board was used during the project, see (Appendix 15.3).

5 LEGAL, SOCIAL AND ETHICAL ISSUES

5.1 LEGAL

5.1.1 GDPR

Due to the application having a login system, personal data about the registered users is stored in the database. The website must comply with the General Data Protection Regulation (GDPR), which came into effect on 25th May 2018 (Intersoft Consulting, 2016). Compliance has been met by ensuring that stored data is completely necessary, and it is only stored for as long as it is required. In addition, information is given to the user as to why their data is stored and how it will be used on the register page. This makes it clear to the user and shows complete transparency. All users' passwords are hashed before being stored in the database and all server requests that are made are processed over HTTPS with SSL encryption.

5.1.2 Cookie Compliance

Cookie compliance is a part of GDPR and means that the website should inform users that the site is using cookies and give them the option to opt in or out of non-essential cookies (GDPR.EU, 2019). This application only uses one cookie, which is essential to maintain a user's logged-in state whilst navigating through pages. A banner is shown to the user when they first open the website. It informs them that cookies are used and for what reason (Figure 3). The banner does not disappear until the user acknowledges the message. This makes the website comply with the EU cookie law and the cookie section in GDPR.

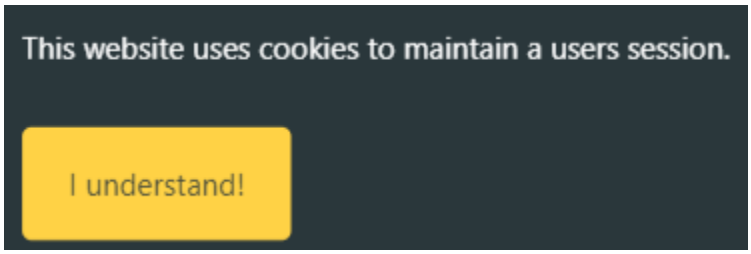


Figure 3. Cookie acknowledgement banner.

5.1.3 Intellectual Property

All videos, images and icons were either designed and created by the developer or were from open source libraries that are free and legal to use for commercial purposes. Many open source libraries were used in both the front-end and middleware of the application that are all free to use.

5.2 Social and Ethical

5.2.1 Web Accessibility Initiative

In 1997, W3C launched the Web Accessibility Initiative (WAI) to help improve the accessibility of the world wide web for people with disabilities (W3C, 2019). The website made in this project has been designed to make it as accessible as possible for those who are visually impaired. This was achieved by using colours for text and backgrounds that have a high contrast and are easily readable. Furthermore, HTML alt tags have been used where necessary to describe what an image is. These tags can be read aloud by browsers to help users who are unable to see the images.

5.2.2 Email Address Use

The use of the users' email addresses has also been assessed and is only used for logging in purposes. This means that the application does not send any marketing or other bulk emails to users that have registered with their email address.

5.2.3 Ethics Application

Before this project started, the university approved an ethics application for requirements gathering and user testing. This project has complied with the university's ethics policy. For testing and verification purposes, participants were required. All participants were made aware of why testing was needed and what they would be doing beforehand. Participation was completely voluntary and could be withdrawn at any point during the sessions.

6 Specification and Design

6.1 Requirements

Requirements were obtained from user stories and were then determined as either part of the minimum viable product or an additional feature. User stories can be seen in (Appendix 15.4).

6.1.1 Functional Requirements

Core Features

Create a progressive web application that:

- Is designed for mobile use and has a responsive and user-friendly interface
- Allows for clients to track their weight
- Allows for instant messaging between users
- Allows for personal trainers to be able to create workouts and assign them to their clients

- Allows for clients to be able to view workouts assigned to them and log what they completed during a workout
- Allows users to be able to see past workouts and their progress made

Desirable Features

- Be able to store files that can be viewed by the client's personal trainer
- Be able to group clients and assign workouts to the group

Optional Features

- Be able to track your nutrition either within the application or by linking to a third-party application
- Personal trainers can advertise to potential new clients who are already signed up to the application
- Be able to import data from a smart device (e.g. smart watch, smart scales)

The requirements that were defined were then expanded upon to turn them into SMART objectives (Appendix 15.5). Using SMART objectives helps keep objectives very well defined by ensuring they are thoroughly thought out. In addition, it helps monitor progress and prioritise work when each goal has a time frame and measurable specifics. A use-case diagram was created to show how users would interact with the functionality of the application. This can be seen in (Appendix 15.14).

6.1.2 Non-functional Requirements

- Application should be compatible with many devices and browsers
- Application should be fast, responsive, and intuitive to use
- Users should have their own login with a secure backend
- Input should be validated on both the client side and server side
- Documentation on how to use the application should be provided

6.2 MODEL-VIEW-CONTROLLER

Model-view-controller (MVC) is a software architectural pattern that is used to separate an application into 3 elements. The pattern, as shown figure 4, was invented by Trygve Reenskaug in 1979, where he described the following 3 elements and what they should represent:

Model

Models are the representation of knowledge. They can either be a single object or a structure of objects. Models are completely independent of the graphical interface and manage the logic of the application.

View

Views are the graphical representation of its corresponding model. It can be used to hide certain attributes of a model as well as show other attributes. This is said to be acting as a presentation filter.

Controller

Controllers are the links between users and the system. They arrange views to present themselves and provide the user with an input. This allows users to give commands or data to the model via the view. Finally, the controller receives outputs and formats them appropriately to then pass on to the corresponding views.

(Reenskaug, 1979)

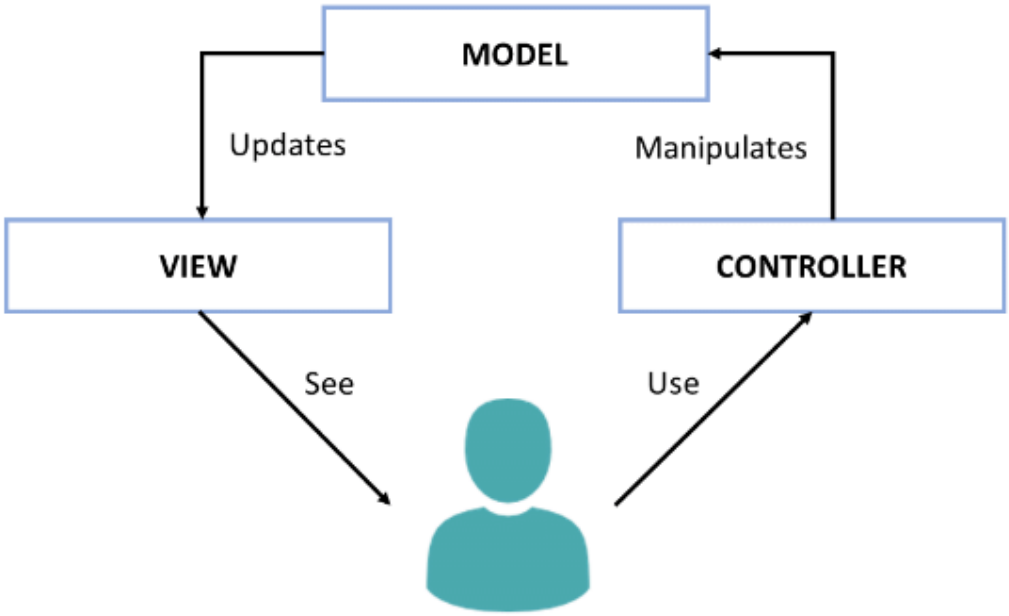


Figure 4. MVC Pattern Diagram. (Nor et al., 2018).

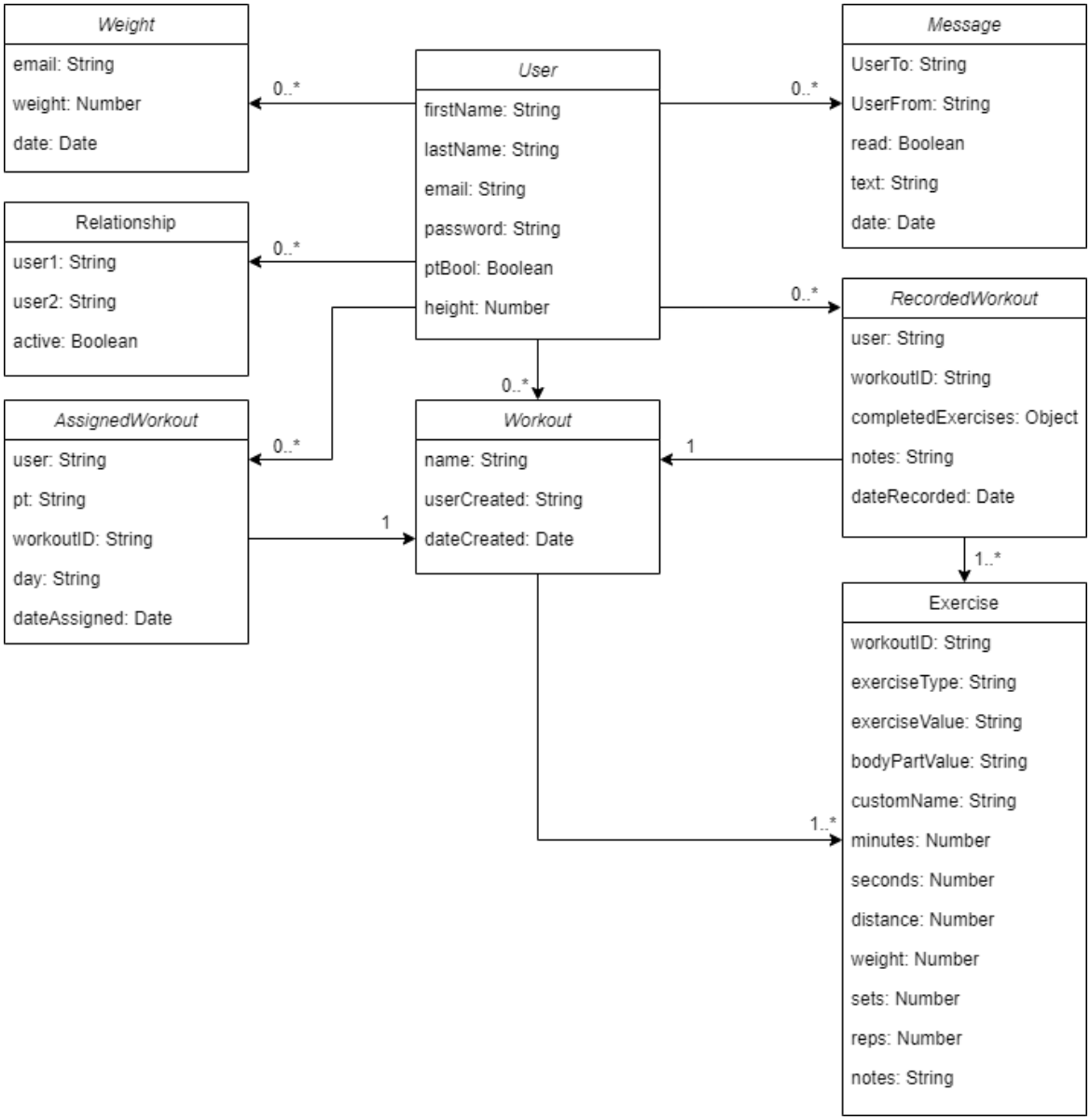
The MVC pattern was used in this project as it allows for the developer to work on the model, view, and controller simultaneously. This helped to be able to focus on building a minimum viable product first and then adding desired features after – this is instead of focusing on one aspect of MVC at once. It is very easy for models have multiple views using the MVC pattern. This was necessary when developing the front-end with ReactJS as it is component-based, and one model can link to multiple components at any one time. In addition, the MVC pattern allows for low coupling as each part of MVC are separate and interactions between each are kept to minimum.

A sequence diagram can be seen in (Appendix 15.15). This shows how the MVC pattern was implemented into this project and the stages that occur within the client, server, and database as the user interacts with the application.

6.3 Database Design

A class diagram was created as a visual representation of how data is stored for the application. It was imported to create this diagram as it is good documentation of the database design and allows other people to understand it. Furthermore, it allows the developer to recognise how changes to the application could affect other areas by looking at how the tables are linked together.

A class diagram was used instead of an entity relationship diagram because a NoSQL database was used, and the data was stored in JSON format. Using a NoSQL database has many advantages, including being able to easily scale the system for future development and allowing for simple implementation.



6.4 User Interface Considerations

User interface designs were made during the initial stage of the project, which can be seen in (Appendix 15.16). When designing the interface, Ben Shneiderman's eight golden rules were kept in mind to help design a user friendly and intuitive application. The rules are:

- Strive for consistency
- Enable frequent users to use shortcuts
- Offer informative feedback
- Design dialogue to yield closure
- Offer simple error handling
- Permit easy reversal of action
- Support internal locus of control
- Reduce short-term memory load

(Shneiderman *et al*., 2009)

To ensure consistency was kept throughout the entire application, a top app bar and bottom navigation bar is visible on every page, this can be seen in figure 5. Having these visible on every page helps the user keep track of where they are within the application, with the ability to quickly change pages or go back to a previous page. Furthermore, the bottom navigation bar encourages users to move around the app and find new features. The Material UI library was used to implement these, as well as the icons, lists and other UI components throughout the application.

Figure 5. Bottom navigation bar and top app bar.

Various shades of green were chosen as the colour scheme for this application, with a contrasting white background. This matched the colour of the logo that was initially designed and kept a satisfactory contrast for visually impaired users. The logo and colour scheme can be seen in figure 6.

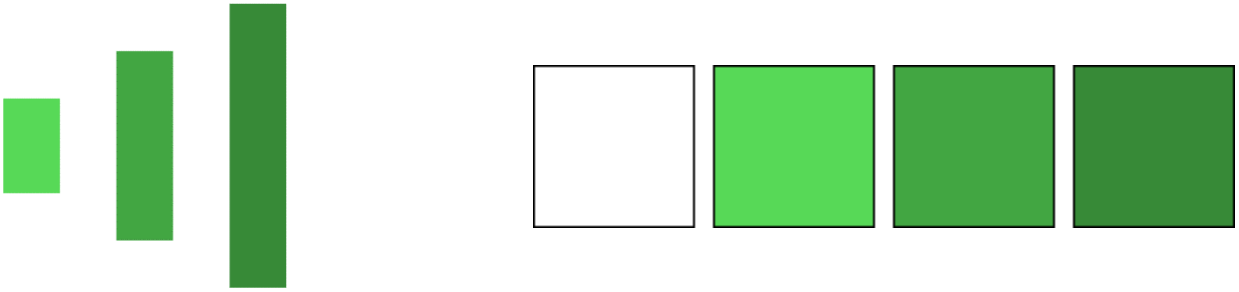


Figure 6. Application logo and colour scheme.

7 DEVELOPMENT TECHNOLOGIES

At the beginning of the project, an investigation was completed to find out which technologies would be most appropriate for developing this application. This can be seen in (Appendix 15.17). Below are the technologies that were chosen and why.

7.1 WEB APPLICATION

For the development of the application, it was decided to create a progressive web application. The application was developed using ReactJS as the front-end framework. A progressive web application was chosen for the following reasons:

- It would decrease development time significantly because only one application would have to be developed which would be compatible on many browsers and devices.
- They can easily be designed to be response and user friendly.
- It is possible to make use of a device's push notification system.
- Service workers can be implemented allowing the application to be used offline.

ReactJS was chosen as the front-end framework for the following reasons:

- It is easy to create a progressive web application
- Lots of free UI libraries that can be imported
- Component based system allows for code to be re-used
- Developer is experienced in JavaScript

7.2 MIDDLEWARE

For the middleware of the application, Node.js was decided to be used. The middleware acts as the server to serve the application, it will also be used as an API for the application endpoints. The API routes are handled by Express. Node.js was chosen because:

- Node.js uses JavaScript which keeps the language consistent.
- Many packages are readily available to be used.
- It is fast and asynchronous.
- It is easily scalable as the application grows.

7.3 DATABASE

For the database storage part of the application, it was decided that a NoSQL database was to be used. The most common NoSQL database, that is also free to use up to 500mb, is MongoDB. A NoSQL database was chosen because:

- It allows for easy scalability and schemas can be changed without affecting the rest of the data in the database as there are no relationships between tables.
- Data can be stored in JSON format, making it much easier to parse.
- There is less maintenance compared to an SQL database.

7.4 HOSTING

For hosting the application, Heroku was chosen as the best option. This was because:

- It is easy to create a microservice to host a ReactJS and Node.js project.
- It can be implemented into a continuous integration and continuous delivery pipeline.
- It is free to use under the 'Free Dyno'

The domain name was also acquired, along with setting up a DNS to provide a secure socket layer (SSL) certificate for the website. For more information about how the website was hosted, please see (Appendix 15.18).

7.5 SECURITY

The security of the application is very important to ensure that personal data is kept safe. Many technologies were used to help protect user data and by making sure that data cannot be accessed unless the user is authenticated. PassportJS was used in this project for user authentication. This is because it implements very easily into a Node.js project and can be used with a library called 'bcyrptjs' to hash all users' passwords before they are stored in the database.

Cookies have been used to keep a users' credentials whilst they are navigating throughout the website. PassportJS uses the cookie to help protect all API routes - if the requests are not coming from an authenticated user then it will get rejected.

A DNS was set up via Cloudflare to ensure the hosted website had SSL encryption. This means that data that is passed via requests to and from the server is encrypted and it would be much harder for someone to operate a man-in-the-middle attack. For a website to meet the progressive web application requirements, it must have an SSL certificate.

8 IMPLEMENTATION

8.1 PRODUCT BACKLOG

A product backlog was created before beginning implementation of the application. This backlog was created by using user stories which were obtained from my client and also included necessary backend requirements. The product backlog was organised by which sprint each objective should be completed in; these will be the deliverables for

each sprint. Each objective in the product backlog is also marked as to whether it is needed for the minimum viable product or if it is an additional feature. This has been used to prioritise work and plan sprints as best as possible. The product backlog can be seen in (Appendix 15.10).

8.2 SPRINTS

8.2.1 Sprint 1

Dates
27/01/2020 - 09/02/2020

Summary
The aim for the first sprint was to complete background research on the issue that the application would solve, complete some initial design work, and set up the first backend features.

Deliverables

- Competitors comparison table
- User stories
- Entity relationship diagram
- UML diagrams
- First design of main pages
- Research into PWA's
- Setup DevOps pipeline
- Setup application hosting on Heroku
- Setup connection to a MongoDB database with CRUD functionality

The planned deliverables for this sprint started by creating the user stories and producing the product backlog for the implementation process. To do this, a short and informal meeting took place with the client to get a better understanding of what the minimum viable product would look like and what additional features could be added.

Background research was carried out to ensure that the application would be feasible and that the right technologies were going to be used. Lots of research was carried out on progressive web applications to ensure that all required features would be possible. A comparison table was created to see what features similar applications had implemented and how this solution would be different. A Sequence diagram, Use-Case diagram and Entity-Relationship diagram were created to show the model of the application and assist with the development. In addition, the initial front-end designs for the application's main pages were created during the first sprint.

A DevOps pipeline was created to be used throughout the implementation stage for continuous integration and deployment. Travis CI is being used to test the application

and push deployment to Heroku, where the application is being hosted. This all happens when a Git push occurs. Basic CRUD functionality was also set up for a MongoDB database at the end of the sprint.

Discussion
Sprint 1 was successful even though there was not any actual development. A lot was learnt from this sprint and helped get a much better understanding of what would take place for the rest of the sprints. Setting up the DevOps pipeline took a little bit longer than expected as Travis CI and Heroku were both new systems to learn. However, this did not have much of an impact on the overall sprint.

8.2.2 Sprint 2

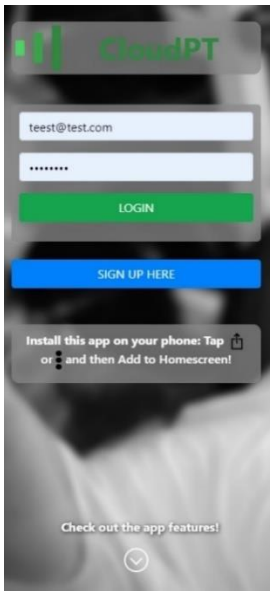
Dates
10/02/20 - 23/02/20

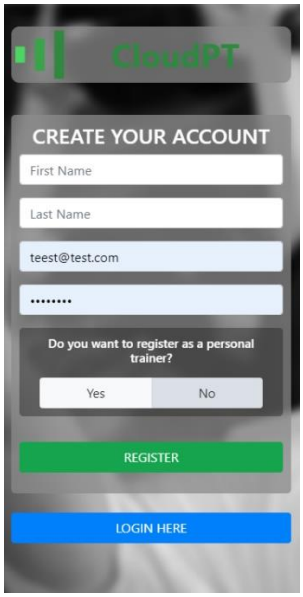
Summary
The plan for Sprint 2 was to properly start development by creating a ReactJS project and implement a working login system. This was alongside front-end login and register forms as well as a basic home page to welcome the user.

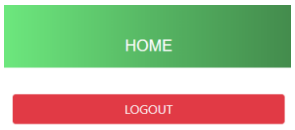
Deliverables
- Setup PassportJS for login and register
- Create login form
- Create register form
- Logout functionality

The first deliverable for this sprint was to create a new ReactJS project and a Node.js backend server with a proxy to allow the two to communicate. The first API endpoints were created to be able to let the front-end communicate to the server.

To implement a login system, PassportJS was used as it integrates very easily into a JavaScript project. PassportJS handles password salting and hashing when a user registers or attempts to login. This allows for the password to be hashed before being stored in the database. Cookies are being used to track a user's sessions and maintain their logged in state while navigating through pages. Logout functionality was also implemented towards the end of the sprint.







Discussion
This sprint was much more challenging than originally thought. This is mainly because ReactJS was a new technology to learn and took longer to understand than planned. However, PassportJS was a familiar package so it was easy to implement into the Node.js server.

An issue did arise about half-way through the sprint when trying to get the react project to communicate to the server due to getting some CORS errors. This was rectified by setting up a proxy for the react project to communicate via.

8.2.3 Sprint 3

Dates
24/02/20 - 08/03/20

Summary
This sprint was used to implement some initial UI designs to help the application feel like the native mobile application. This would be achieved by learning a bit more in depth about ReactJS and any UI libraries that might be needed to achieve this. Furthermore, it was used to implement functionality to allow users to add and change their personal details.

Deliverables
- Create Logo
- Create Brochure Submission
- Implement mobile UI
- Allow users to store personal details

The deliverables for this sprint started with implementing a mobile friendly UI. To achieve this, a lot of time was spent on researching some UI libraries to find one that would fit best. Material UI was chosen and was then used to implement a bottom navigation bar as well as a top app bar.

Another deliverable for this sprint was to allow users to input and change their personal information. This meant that clients were able to input their weight and height which they would be able to see on a graph using ChartJS. In addition, some basic functionality so that all users where able to change their name, email, and password. As part of this sprint it was also necessary that a logo was produced, alongside a submission for the project showcase brochure (Appendix 15.8).

Discussion
Overall, this sprint went well, and all deliverables were met on time. A much better understand of what React components were going to be used was grasped and how a consistent UI design was going to be kept throughout the application.

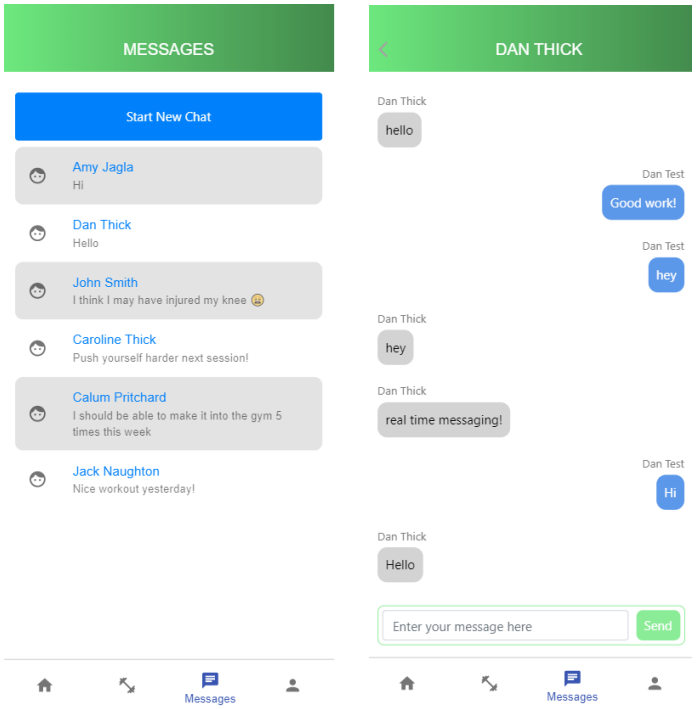
8.2.4 Sprint 4

Dates
09/03/20 - 29/03/20

Summary
The aim for this sprint was to implement instant messaging into the application using WebSockets. The plan for this was for users to be able to start a conversation with other users via their email address and to implement WebSockets to allow for real time updates.

Deliverables
- Messaging between users
- WebSocket implementation

To allow for users to message each other, the first part was to implement the front-end designs for the chat layout and list of messages. Once this was completed, functionality to start a new chat with an existing user was made along with the ability to send messages that are stored in the database. Towards the end of the second week of the sprint, implementation of WebSockets took place to allow for real-time messaging between users.



Discussion

This sprint turned out to be much more time consuming than originally expected. WebSockets was a new technology to learn and the sprint plan did not include any time to learn it, although it should have. This resulted in more hours being put in than planned towards the end of the sprint to ensure that the deliverables were completed on time. Implementing the front end and the API endpoints was simple and were kept to the schedule as planned.

8.2.5 Sprint 5

Dates

30/03/20 - 12/04/20

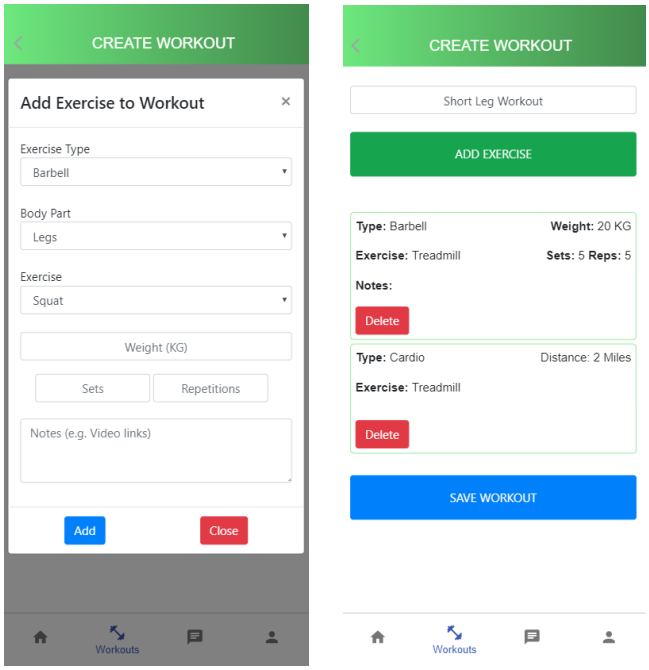
Summary

For this sprint to be achieved, functionality for personal trainers to be able to fully create and save custom workouts needed to be implemented. Personal trainers then need to be able to assign any workout that they have created to their clients, for them to be completed on a specific day.

Deliverables

- Workout creation
- Add exercises to workouts
- Assign workouts to clients

The deliverables for this sprint were for personal trainers to be able to create workouts and add any number of exercises to the workouts. A workout should be able to be saved, which will later be used to assign a workout to a client.



Discussion

Sprint 5 was very successful, and all of the deliverables were completed on time. Creating the page to add exercises needed more design time as it required a lot of user input and was challenging to make it as simple as possible for the user. However, the extra time spent designing was worth it as now the entire workout creation process is very simple and works well.

8.2.6 Sprint 6

Dates

13/04/20 - 26/04/20

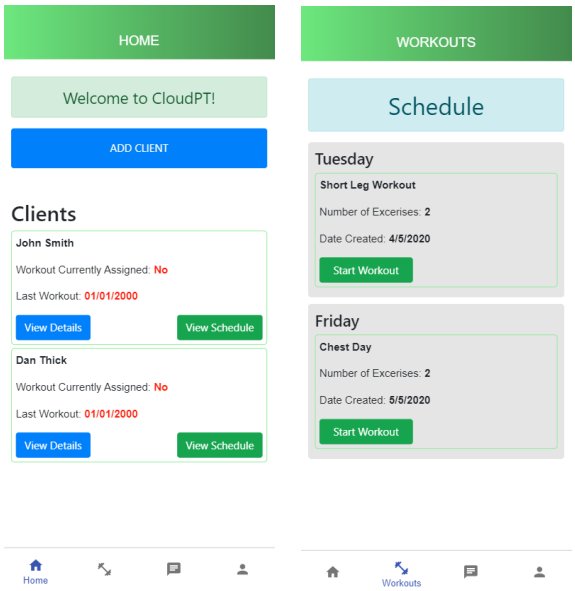
Summary

The aim of this sprint was to allow for personal trainers to add new clients and view a list of all their clients. As well as this, the plan was to add functionality to allow clients to be able to view their workouts with a schedule broken down day by day.

Deliverables

- Ability to add clients
- View list of clients
- Clients can view their workout schedule
- Create poster

The deliverables for this sprint were to implement functionality to allow personal trainers to add clients via their email address. Personal trainers should then be able to view a list of all their clients on the application homepage, each with a button to view details about the client and view their schedule. For clients, their workout page will be filled with workouts that are assigned to them and what day they should be completed.



The last deliverable for this sprint was to create a poster for the project showcase. The poster was used to advertise the project to colleagues, employees, and university staff (Appendix 15.9).

Discussion

The majority of the deliverables were completed for this sprint, but due to the time taken to complete them, a deliverable to allow clients to record their workout had to be pushed back to sprint 7. This meant that there was a significant amount of work left to complete in the last sprint. Despite this, a lot of functionality was implemented throughout sprint 6 as well as creating the project showcase poster.

8.2.7 Sprint 7

Dates

27/04/20 - 10/05/20

Summary

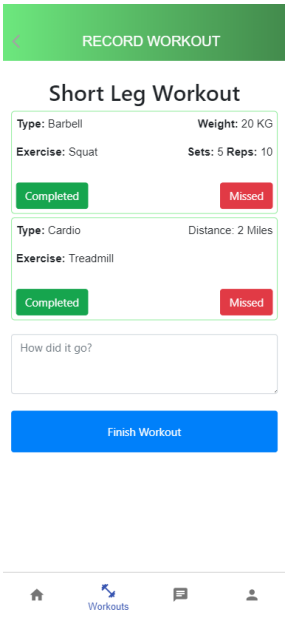
For the last sprint, a small amount of development needed to be done in order to complete functionality of the application. In addition to this, some final user testing needed to be conducted on the finished application.

Deliverables

- Clients can record their workouts
- View workout history
- Implement service workers
- User testing

The deliverables required for this sprint included adding functionality so that clients were able to record their workouts. This data then needed to be used to allow personal trainers and clients to be able to see their workout history. Service workers then needed to be implemented into the application to allow for the application to be useable offline

to an extent. Service workers are needed for progressive web applications so that pages can be cached locally. The last deliverable for this sprint was to carry out some final user testing on the completed application.



Discussion

More had to be completed in this sprint than was planned due to other sprints over-running and deliverables being delayed. Despite this, all the deliverables were met, and the sprint was a success. All development was completed in the first week of the sprint which meant that a whole week was dedicated to user testing.

9 TESTING

9.1 FUNCTIONALITY TESTING

Functional testing is used to validate that the functions throughout the application are working as expected. It is a form of black-box testing, which means that the internals of the system are unknown, and the only focus is the output generated by an input. Functional testing is used to make sure that the application fulfils the user's requirements (Gao, H -S. J Tsao and Ye Wu, 2003).

User stories were used help create the plan for the functional tests. Towards the end of each sprint, functional testing took place on features that were implemented during that sprint. All possible scenarios were tested to ensure that the actual outcome of the test matches the expected outcome. Completing this testing ensured that all of the user's requirements were met and the application behaved as expected. For the complete functionality testing that took place, please see (Appendix 15.20).

9.2 Usability Testing

Usability testing was important to carry out because it helped the developer get a better understanding as to how users interact with the application. It was used to see issues there were with the both the applications design and functionality, as well as learning what users did and did not like about it. Four students from the University of Plymouth, who studied different courses, including the project client, participated in the usability testing. They were asked to complete several core tasks throughout the application and give feedback about the process.

Due to the outbreak of COVID-19, it became impossible to conduct face-to-face usability testing. This meant that remote usability testing had to take place instead. Due to this, it was not possible to watch how users were interacting with the system or see what tasks they found difficult. However, this did result in it being an unmoderated test, meaning that the developer was not present during testing and could not help with any issues. This created a better real-life scenario for testing to take place.

After each user had completed all of the required tasks, they were then asked to anonymously complete a survey. The survey (Appendix 15.21) consisted of questions about how easy tasks were to complete as well as any changes that the user would make.

Overall, the feedback from the usability testing was very positive from all participants. Some design changes were made to make the readability of information easier. A ‘card’ system was implemented to help split information and list items. For more information about the usability testing that took place and changes that were made, please see (Appendix 15.21).

9.3 Unit Testing

As discussed about previously, the agile framework: test-driven development was used during the development process. TDD helped significantly reduce the number of bugs throughout development, as well as finding and fixing them much quicker than normal. Unit testing is when individual components of the software are tested to validate that they are working correctly.

To conduct unit tests on the API, Mocha (Mocha, 2019) and Chai (Chai, 2018) were the two libraries that were used. Mocha allows for tests to be organised by creating groups of tests that run in a specific order. Chai allows for functions that send requests to the API and confirms their responses by using assertions. Jest (Jest, 2017) and Enzyme (Enzyme, 2020) are the libraries that were used to create unit tests for the ReactJS front-end of the application. These worked in a similar way to Mocha and Chai. Jest provided the organisation of the tests, whilst Enzyme provided functions to successfully test individual ReactJS components. For more information about the unit tests that were implemented, please see (Appendix 15.22).

10 End Project Report

10.1 Project Summary

The overall aim of this project was to create a web application that helped personal trainers keep track of their client's workouts and progress, whilst being able to communicate with them. The required features were successfully implemented into the application and the client was very happy with the application, making it a success.

10.2 Project Objectives Review

All of the project objectives were met to a satisfactory level, details of this can be seen below.

10.2.1 Identify problem and complete market research

This objective was met by first identifying the problem by discussing it with my client who has first-hand experience with the issue. Market research was then completed by first looking at the current solutions that were available on the market. This helped to understand what features were currently being implemented to help solve the issue, which in turn gave a better understanding as to what would make this application stand out.

10.2.2 Analyse user stories to gather requirements for the application

User stories were thought of whilst discussing with the client what they wanted from the application. These user stories included what the personal trainer would want to do, as well as what the personal trainer's clients would want to be able to do. The user stories were then used to gather and finalise the requirements of the application before development began. Finally, the requirements were then organised in to two parts: the minimum viable product and preferred or desirable features.

10.2.3 Setup continuous integration and continuous delivery

A continuous integration and continuous delivery pipeline were set up before any development began. This was achieved using Travis CI and Heroku. Overall, this greatly reduced time spent running tests and deploying the application after features had been added due to it all being automatic.

10.2.4 Design and implement a database schema

A database schema was designed during the project initiation. This was then later implemented using MongoDB as a NoSQL database. The database had full create, read, update, and delete (CRUD) functionality and was fully utilised throughout the entire mobile application.

10.2.5 Design and implement an API/middleware

All required API routes were considered before beginning the implementation of the API, these can be seen in (Appendix 15.19). The middleware successfully connects the web application to the database through an API. It also handles all of the user authentication and protection of users' data.

10.2.6 Design and implement a progressive web application

Before the implementation of the application, an investigation took place into the user interface and any HCI considerations that should be made. Mock-ups of what the interface was going to look like were drawn to help visualise the plan of the application. Furthermore, a sequence diagram was produced to show how the user interacting with the mobile application would affect the system as a whole. The model-view-controller pattern was used to implement the application successfully. The application also met all of the requirements of a progressive web application and can be installed on a wide range of devices.

10.3 Project Changes

All of the technologies that were chosen from the initial investigation remained the same throughout the entire project. However, due to ReactJS being a new technology for the developer, more time had to be set aside than expected at the beginning of the project to learn it.

One of the original features that was discussed, was to implement a calendar that the personal trainers would be able to use. This was quickly removed as a requirement after research took place into how the implementation would take place. The client decided that the functionality of a calendar within the application would be limited compared to readily available ones. Despite this, all other requirements were met, and the project did not fall behind schedule.

11 Post-Mortem

11.1 Project Objectives Evaluation

All of the required objectives were met throughout the project, making it a success. The required objectives covered everything to make a suitable application that solved the identified problem. The focus of the objectives was to help personal trainers communicate with their online clients through being able to assign workouts and track their progress. Having a real client massively helped when it came to define the project objectives and then the application requirements.

All of the core requirements were met during the implementation of the application. However, none of the desired or optional requirements that were listed during the start

of the project were implemented. This was mainly due to time constraints on the project, which was further emphasised because of having to learn many new technologies.

11.2 Development Process Evaluation

An agile approach was taken during the development process which helped the project stay on track and ensure that requirements were met on time. It allowed the developer to have some flexibility with the plan for development. This was useful as the developer had to learn many new technologies throughout the development process of the application, which took more time than was originally planned. An example of this was learning ReactJS. In the project plan, very little time was set aside to learn how to use components and other React features. Despite this, the plan that can be seen in (Appendix 15.11), was kept to very closely during development. Apart from towards the final two sprints where some changes had to be made due to tasks taking longer, this was discussed in the sprint discussions.

Using GitHub for version control was very helpful during development as it allowed the developer to easily keep track of what changes had been made to the application and when. This meant that when conducting a sprint review at the end of each one, it was possible to go back and check the log of commits that had been made to the repository to see when features had been implemented. Furthermore, it also gave a sense of confidence to the developer as it allowed for experimental changes to be made with the possibility of rolling back.

Testing was a big part of development and helped tremendously with ensuring that functionality was correct, and the user interface was intuitive. The project client would have the chance to test the application after each sprint and provide feedback. However, it would have been useful if more user testing took place during development as this would have given a wider range of feedback to help further improve the application.

11.3 Project Management Evaluation

Scrum, an agile framework, was used to help manage the project. It meant that the project could be broken down into two-week sprints and that it was available to test after each sprint. This resulted in feedback being provided more frequently than normal and further helped the development of the application. To keep track of sprints and the product backlog, Trello was used. Trello allowed all requirements to have deadlines, checklists, and labels. It helped keep track of sprints and ensured that all requirements were met. In addition to this, meetings took place with a project supervisor every 2-weeks to discuss progress and any issues that have been encountered. This project has shown how important it is to create a project plan beforehand and continuing to follow it throughout.

11.4 Technologies Evaluation

The technologies that were chosen during the initial investigation were appropriate for this project, and no major issues occurred during development. Choosing popular languages and frameworks meant that there was plenty of support online to help for any small issues that arose. Some JavaScript libraries that were used did have poor documentation and it took longer to implement due to that.

More time should have been spent investigating the difference between ReactJS and React Native because as development progressed, more was learnt about both technologies. React Native would have been a better technology to implement a progressive web application as it has a larger support for mobile UI libraries. React Native was disregarded due to the steep learning curve that comes with it. However, reflecting on this, learning React Native would have saved time in the long run.

11.5 Developer Performance Evaluation

Throughout the project, productivity levels were kept high. This was because I found the project interesting and thoroughly enjoyed developing it. Although I did have concerns at the start of the project about how steep the learning curve was going to be for some of these new technologies, I was able to meet all deadlines and produced work of high quality.

Over the course of the project, I have learnt many new skills in both project management and software development. I have learnt how to use an agile methodology with a scrum framework to manage an entire project from start to finish. I have also learnt how to develop an application using the MERN stack, whilst learning ReactJS from scratch. These skills will carry over to working in industry and will help my performance outside of university. I am very proud of the application that was produced and also believe that it does have real-world applications. Furthermore, this project has helped me get a better understand as to what I would like to pursue doing for a career.

12 Conclusion

The aim of this project was to help personal trainers with their online client base. This has been achieved by meeting all the requirements and objectives that were set out. Developing CloudPT has been a huge success and I am very proud of the product that has been created. The application has been greatly praised by the project client and other users who have tested it. Although at times being challenging as it is the largest project that I have ever completed, it was managed very well, and a lot was learnt from the process. My plan is to further develop this application, with more features being added in the future to improve the user experience.

13 FURTHER DEVELOPMENT

This application could prove to be very popular for personal trainers to use with their clients. Throughout developing this application, I have learnt a lot of new project management and technological skills. This has helped me understand areas of the application that would benefit with further development. Currently, the application requests data very often and the data is not stored in cached memory which makes offline use very limited. With some additional development, local storage could be utilised to improve the application loading times and improve offline functionality.

Further features could be implemented to help make the application more useful. Whilst conducting a final meeting with my client, it was discussed that when a user records their workout, it would be better if they could specifically record what they completed. For example, how many sets and repetitions were completed compared to what should have been completed. This would give the personal trainer a much more in-depth view of their client's workout. Additionally, the use of graphs and charts could be developed further to make viewing users' workout data a more user-friendly experience.

Furthermore, features discussed with the client that were then decided upon being optional features were not implemented during this project. These features were:

- Ability to store files for personal trainers to see (e.g. health records)
- Be able to group clients and assign programmes to groups of clients
- Be able to track nutrition either within the application or using a third-party application
- Ability to link smart devices and use its data
- Ability for personal trainers to advertise to new clients

Implementing these features could greatly improve the overall use of the application and would make it a much more helpful application for personal trainers and their clients.

14 REFERENCES

1. My PT Hub. (2020). My PT Hub - The World's Number One Personal Trainer Software. [online] Available at: https://www.mypthub.net/ [Accessed 23 Apr. 2020].

2. TrueCoach (2020). Personalized Coaching Made Easy. [online] TrueCoach. Available at: https://truecoach.co/ [Accessed 30 Apr. 2020].

3. Rutgers, H., Hollasch, K., Ludwig, S., Gausselmann, S., Rump, C. and Seelemeyer, L. (2020). EuropeActive European Health & Fitness Market Report 2020. EuropeActive.

4. Future Fit Training (2019). 15 Personal Training Employment Statistics. [online] Future Fit Training. Available at: https://www.futurefit.co.uk/content-hub/15-personal-training-employment-statistics/ [Accessed 26 May 2020].

5. Facebook. (2020). Facebook. [online] Available at: https://facebook.com [Accessed 7 May 2020].

6. Instagram. (2020). Instagram. [online] Available at: https://instagram.com [Accessed 7 May 2020].

7. Google. (2020). Google Sheets: Free Online Spreadsheets for Personal Use. [online] Available at: https://www.google.co.uk/sheets/about/ [Accessed 7 May 2020].

8. Saye, T. (2018). The Future of Online Fitness. [online] www.ptdistinction.com. Available at: https://www.ptdistinction.com/blog/the-future-of-online-fitness [Accessed 7 May 2020].

9. Urbanek, E. (2017). Social Media and the Fitness Industry: Statistics You Need to Know. [online] Simplestrat.com. Available at: https://blog.simplestrat.com/social-media-and-the-fitness-industry-statistics [Accessed 7 May 2020].

10. web.dev. (2020). Progressive Web Apps. [online] Available at: https://web.dev/progressive-web-apps/ [Accessed 7 May 2020].

11. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D. (2001). Manifesto for Agile Software Development. [online] Available at: https://agilemanifesto.org/ [Accessed 12 May 2020].

12. Pavel Kukhnavets (2019). How to run Scrum efficiently in 2019? Quick guide for beginners. [online] Habr.com. Available at: https://habr.com/en/company/hygger/blog/455022/ [Accessed 13 May 2020].

13. Atlassian (2018). Scrum - what it is, how it works, and why it is awesome. [online] Atlassian. Available at: https://www.atlassian.com/agile/scrum [Accessed 13 May 2020].

14. GitHub (2020). Build software better, together. [online] GitHub. Available at: https://github.com/ [Accessed 15 May 2020].

15. Trello (2020b). Trello. [online] @trello. Available at: https://trello.com/ [Accessed 15 May 2020].

16. Travis CI (2020c). Travis CI - Test and Deploy with Confidence. [online] Travis-ci.com. Available at: https://travis-ci.com/ [Accessed 16 May 2020].

17. Heroku (2020b). Cloud Application Platform | Heroku. [online] Heroku.com. Available at: https://www.heroku.com/ [Accessed 25 Jan. 2020].

18. GDPR.EU (2019a). Cookies, the GDPR, and the ePrivacy Directive - GDPR.eu. [online] GDPR.eu. Available at: https://gdpr.eu/cookies/ [Accessed 17 May 2020].

19. W3C (2019). Home. [online] Web Accessibility Initiative (WAI). Available at: https://www.w3.org/WAI/ [Accessed 17 May 2020].

20. Intersoft Consulting (2016). General Data Protection Regulation (GDPR). [online] General Data Protection Regulation (GDPR). Available at: https://gdpr-info.eu/ [Accessed 17 May 2020].

21. Reenskaug, T. (1979). The original MVC reports. [online] Available at: https://pdfs.semanticscholar.org/81f8/5bc041f48f33c67a8362c643cae8d9173eee.pdf [Accessed 18 May 2020].

22. Nor, R., Jalaldeen, M., Razi, M., Zakaria, A., Safiuddin, A., Fakhri, A., Zulaiha, P. and Saat, A. (2018). Cloudemy: Step into the Cloud. [online] Available at: https://journal.utem.edu.my/index.php/jtec/article/viewFile/2975/2107 [Accessed 18 May 2020].

23. Shneiderman, B., Plaisant, C., Cohen, M. and Jacobs, S. (2009). Designing the user interface: strategies for effective human-computer interaction. 5th ed. Boston Pearson.

24. Gao, J., H -S. J Tsao and Ye Wu (2003). Testing and quality assurance for component-based software. [online] Boston, Mass.: Artech House. Available at: https://books.google.co.uk/books?id=VoCX09hOsCoC&pg=PA170&redir_esc=y#v=onepage&q=black&f=false [Accessed 25 May 2020].

25. Chai (2018). Chai. [online] Chaijs.com. Available at: https://www.chaijs.com/ [Accessed 26 May 2020].

26. Mocha (2019c). Mocha - the fun, simple, flexible JavaScript test framework. [online] Mochajs.org. Available at: https://mochajs.org/ [Accessed 26 May 2020].

27. Jest (2017). Jest · Delightful JavaScript Testing. [online] Jestjs.io. Available at: https://jestjs.io/en/ [Accessed 26 May 2020].

28. Enzyme (2020a). Introduction · Enzyme. [online] enzymejs.github.io. Available at: https://enzymejs.github.io/enzyme/ [Accessed 26 May 2020].

15 APPENDICES

15.1 INSTALLATION GUIDE

To run the application, first ensure that Node.js is installed on your machine. This can be found here:
https://nodejs.org/en/download/

Then run the following commands in a terminal:

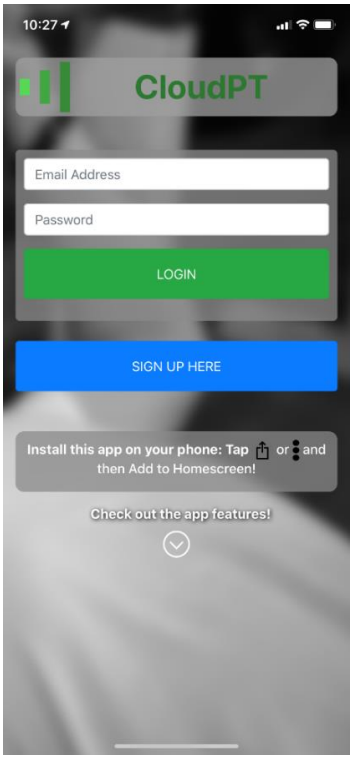
```
git clone https://github.com/danthick/PRCO304.git
cd prco304
npm install
npm start
```

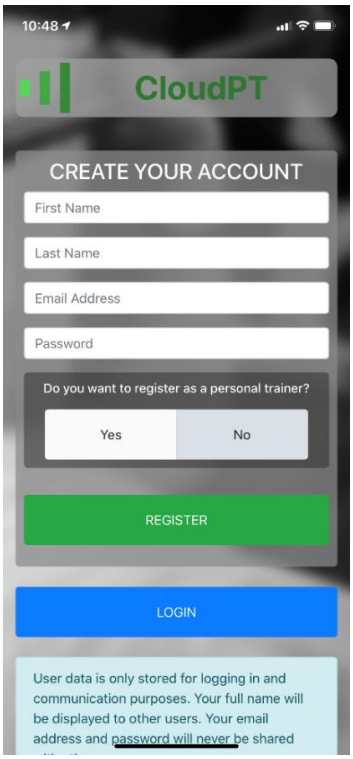
The application will then be running on port 3000, and can be accessed at:
http://localhost:3000

15.2 USER GUIDE

15.2.1 Register and Login

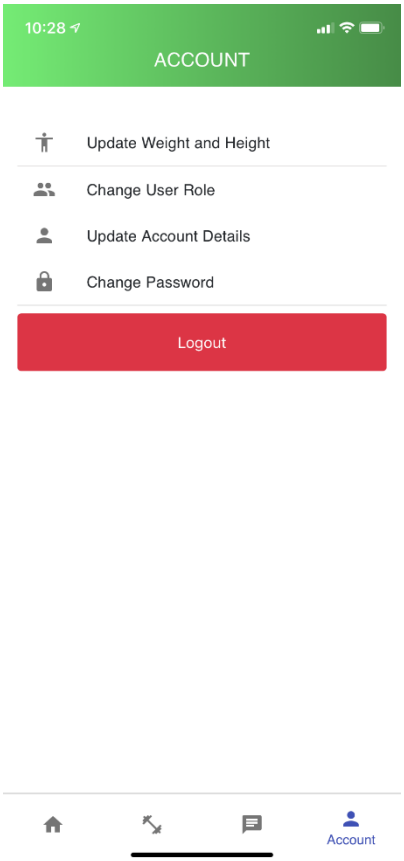
On the homepage of the application a login page is presented. If the you do not have an account click ‘Sign Up Here’, this will navigate you to a register page where you will be able to enter your details and either register as a personal trainer or client. You will then be re-directed to login once registration is complete.

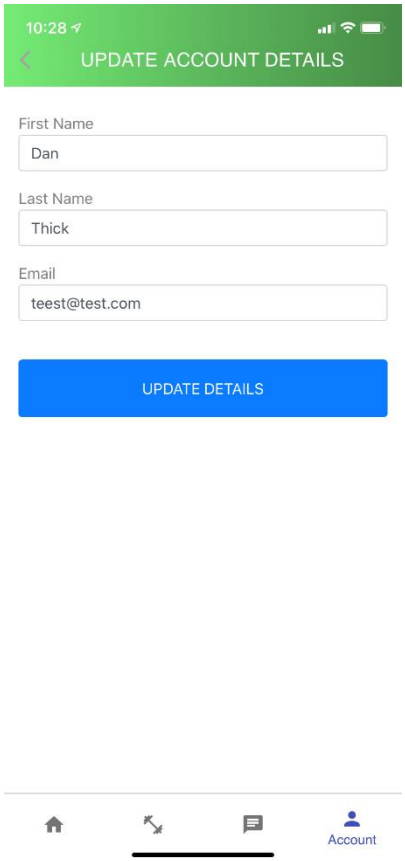


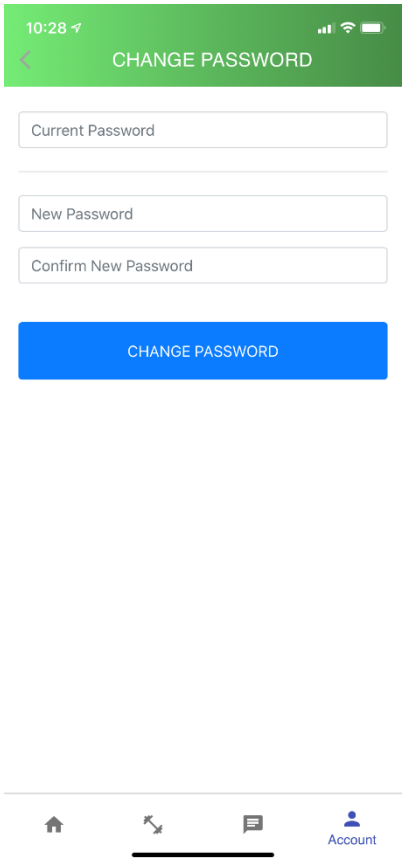


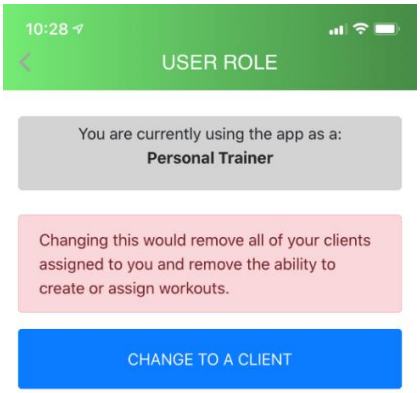
15.2.2 Update User Information

To update any account information, navigate to the ‘Account’ page. From there you are able to either change your password or update your account details. These details are then saved when clicking the blue bottoms to confirm the changes. Furthermore, you are able to change your user role between a client and personal trainer. Please bear in mind that formation such as, created workouts, recorded workouts and clients will be lost.



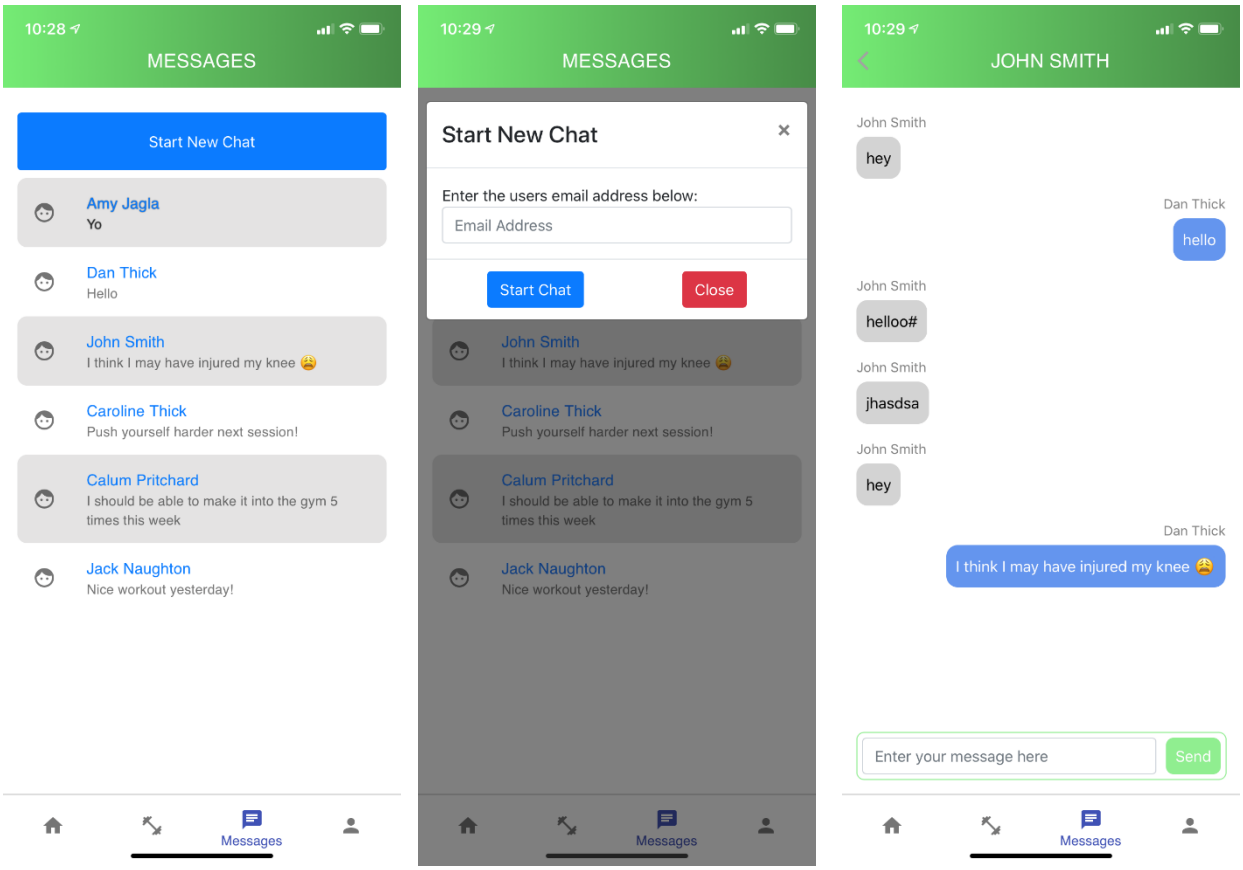






15.2.3 Sending Messages

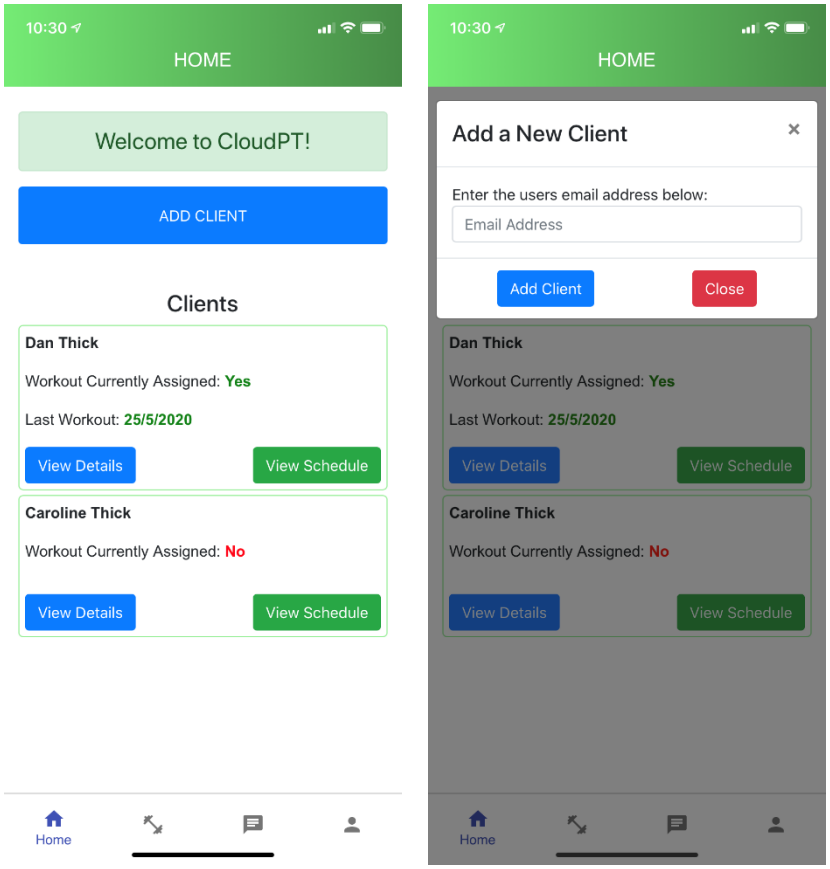
To send a message, first navigate to the messages page of the application. From there you can either select a previous conversation or start a new chat. To start a new chat, click ‘Start new chat’. From there you are able to enter a user’s email address and then click, ‘Start Chat’. The chat window will then open and messages can be sent by inputting text and clicking the ‘Send’ button.



15.2.4 Adding a Client

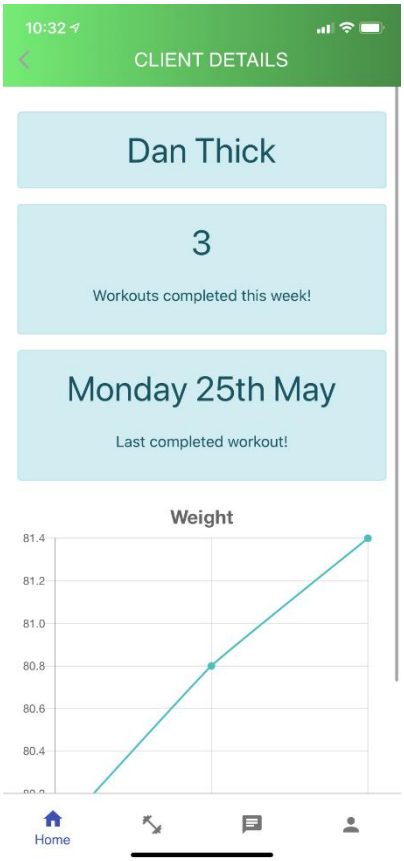
If you are registered as a personal trainer then you can add clients to your account. This will link a client and you can assign workouts, see their details, and view workout history.

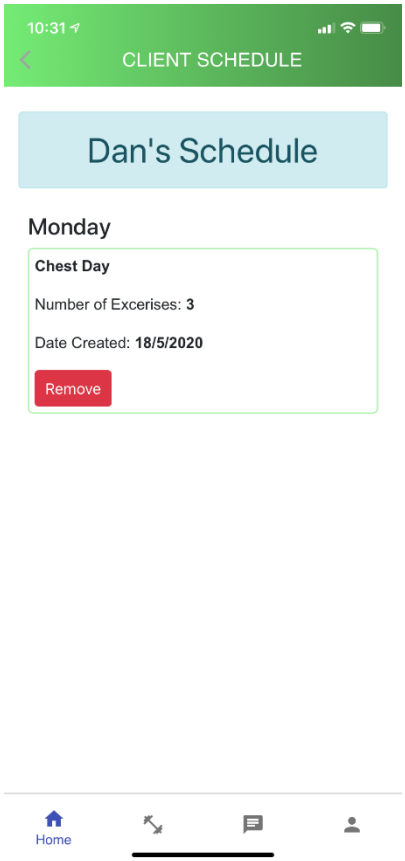
To add a client, navigate to the homepage of the application. From there you can view a list of your clients or add a new one. Click ‘Add Client’ and input their email address into the popup window. Finally, click ‘Add Client’ and the client will be added and will show in the list.



15.2.5 Viewing Client Information

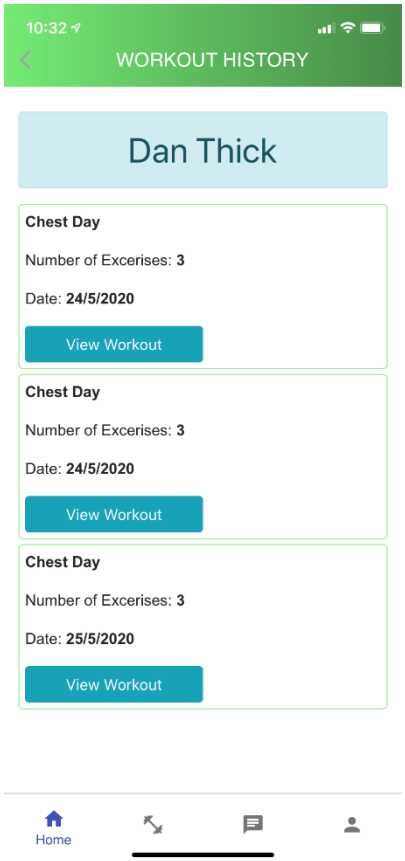
To view a client's information, first navigate to the home page of the application. From there click 'View Details'. Here you are presented with some information about the client's workouts and their weight. You can remove a client or view their workout history from here also.

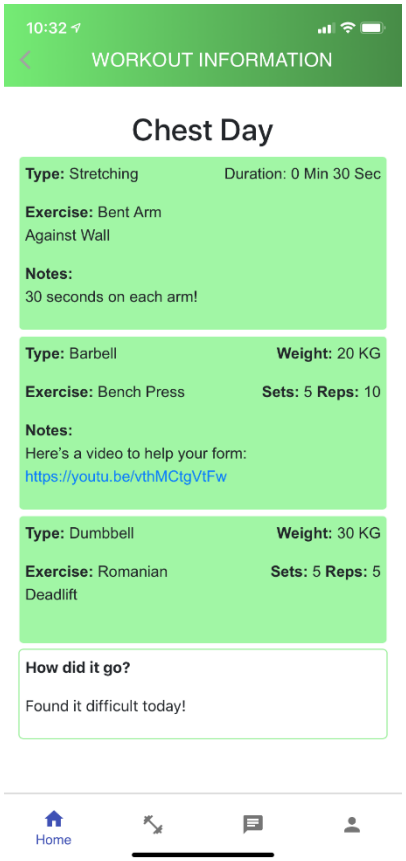




15.2.6 Viewing Client Workout History

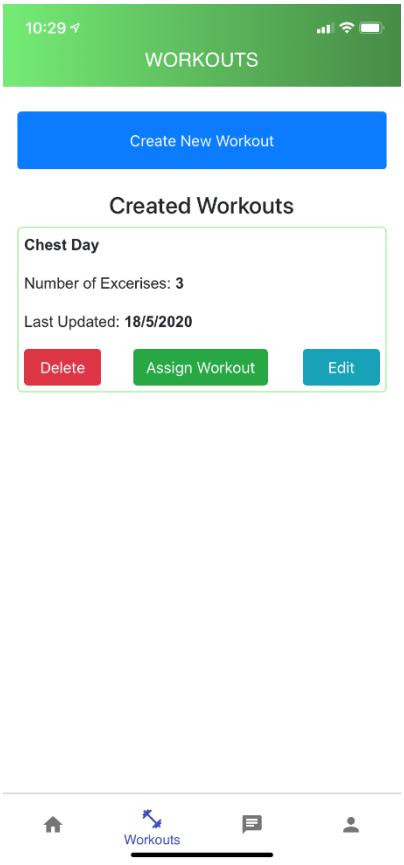
To view a client's workout history, navigate to the homepage and click 'View Details' for a specific client. Then click 'View Workout History'. A list will be displayed of all the workouts that have been completed by that client. You can click 'View Workout' to view how the workout went, along with any notes that were added to the workout. Exercises with either be marked in green or red depending on whether the exercise was completed on missed.

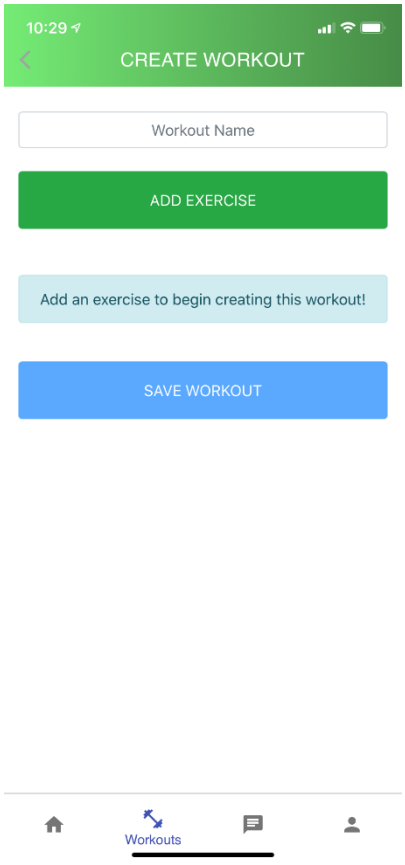


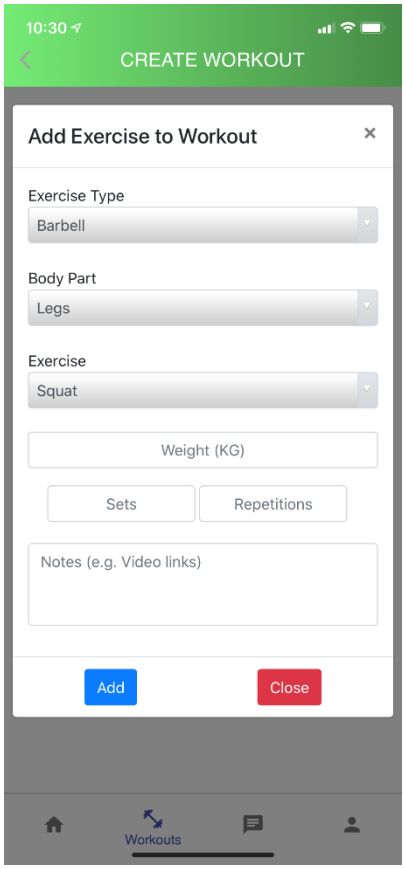


15.2.7 Creating a Workout

To create a new work up, first navigate to the workout page of the application. From there you will see a list of your previously created workouts with the ability to, delete, assign, and edit. To create a new workout, click ‘Create New Workout’. You will then be able to enter your workout name along with the ability to add exercises to the workout. Click ‘Add Exercise’ to be able to input details about it. Once the exercise has been added, it will show in a list and you will be able to save the workout. Workouts can have as many exercises as you would like.

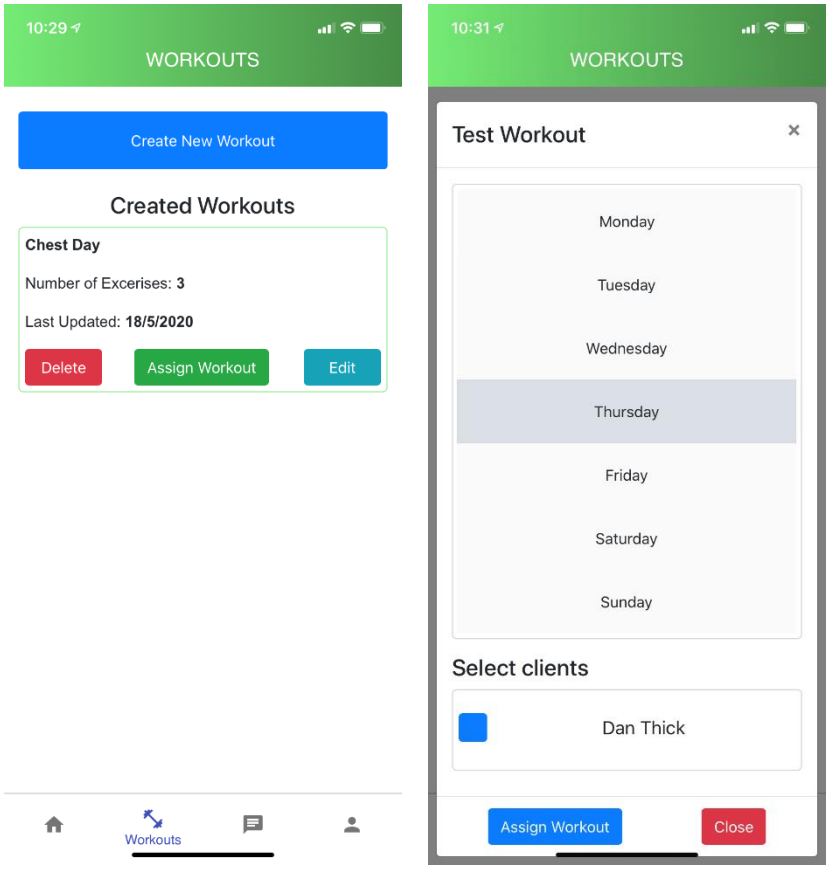






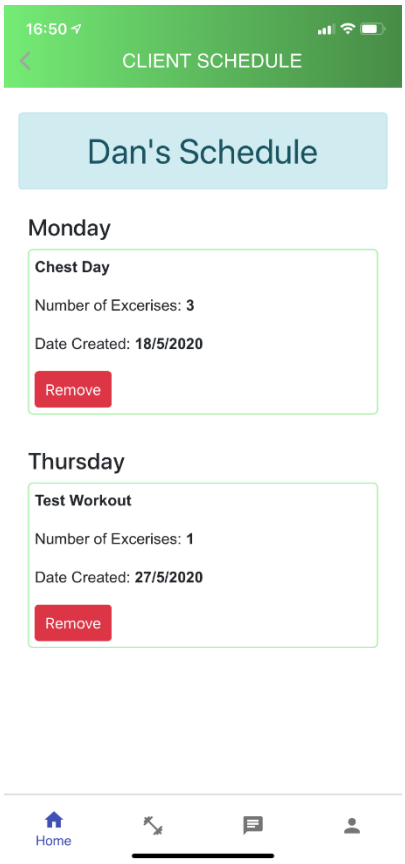
15.2.8 Assigning a Workout

To assign a workout to a client, first navigate to the workouts page of the application. From there, click ‘Assign’ on a specific workout of your choice. A popup window will appear with the option of what day you would like to assign the workout to and to which clients. You are able to select multiple clients if you wish.



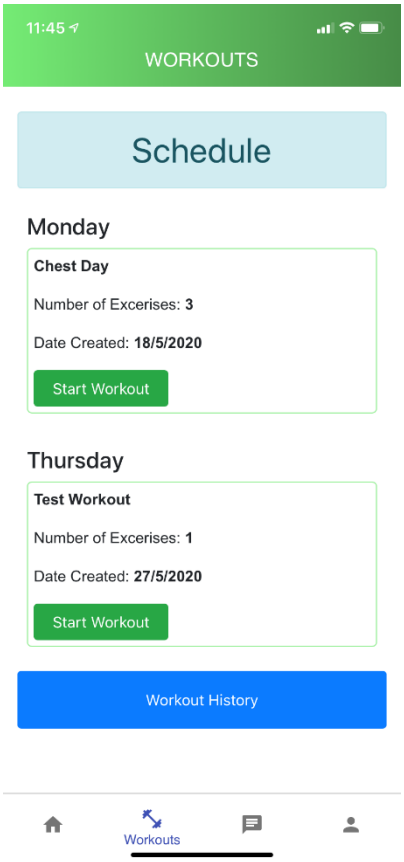
15.2.9 View Schedule

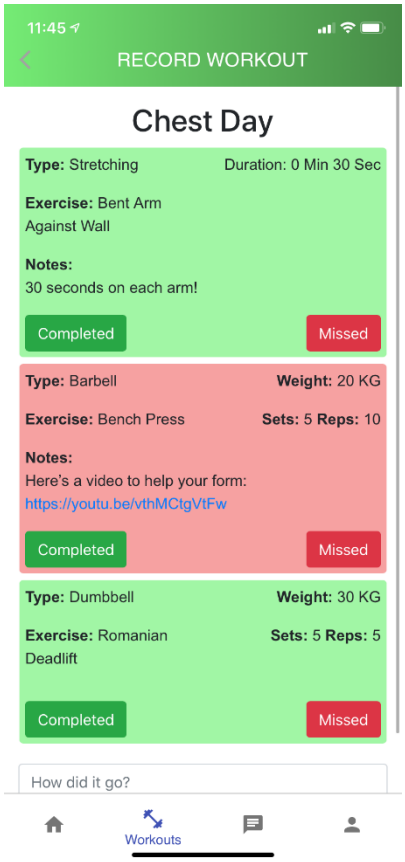
To view a client's schedule, first navigate to the homepage of the application. From there, click 'View Schedule' on one of the clients in the list. You will then be presented with the client's schedule broken down into each day. You are able to see what workouts have been assigned them, with the ability to remove them.

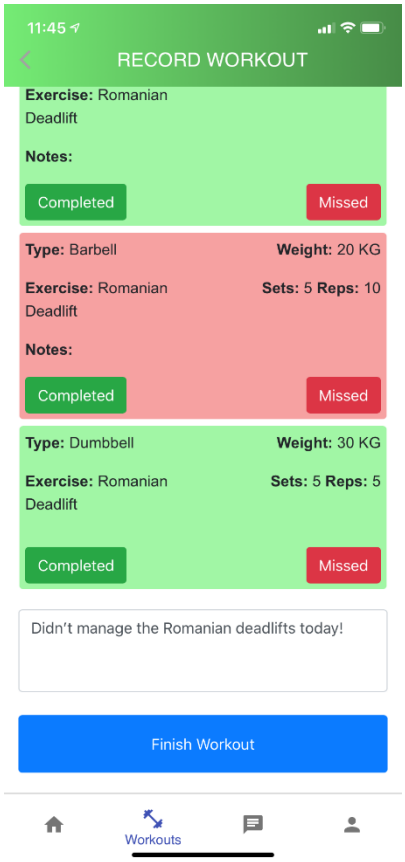


15.2.10 Record Workout

To record a workout, first navigate to the workouts page of the application. From there, click ‘Start Workout’ on the specific workout. Once the workout has started, you are able to mark each exercise as either completed or missed. You can also add any notes that you want to at the bottom of the page. Then click ‘Finish Workout’ and the workout will be saved. This will then be visible to you and your personal trainer in the workout history section.

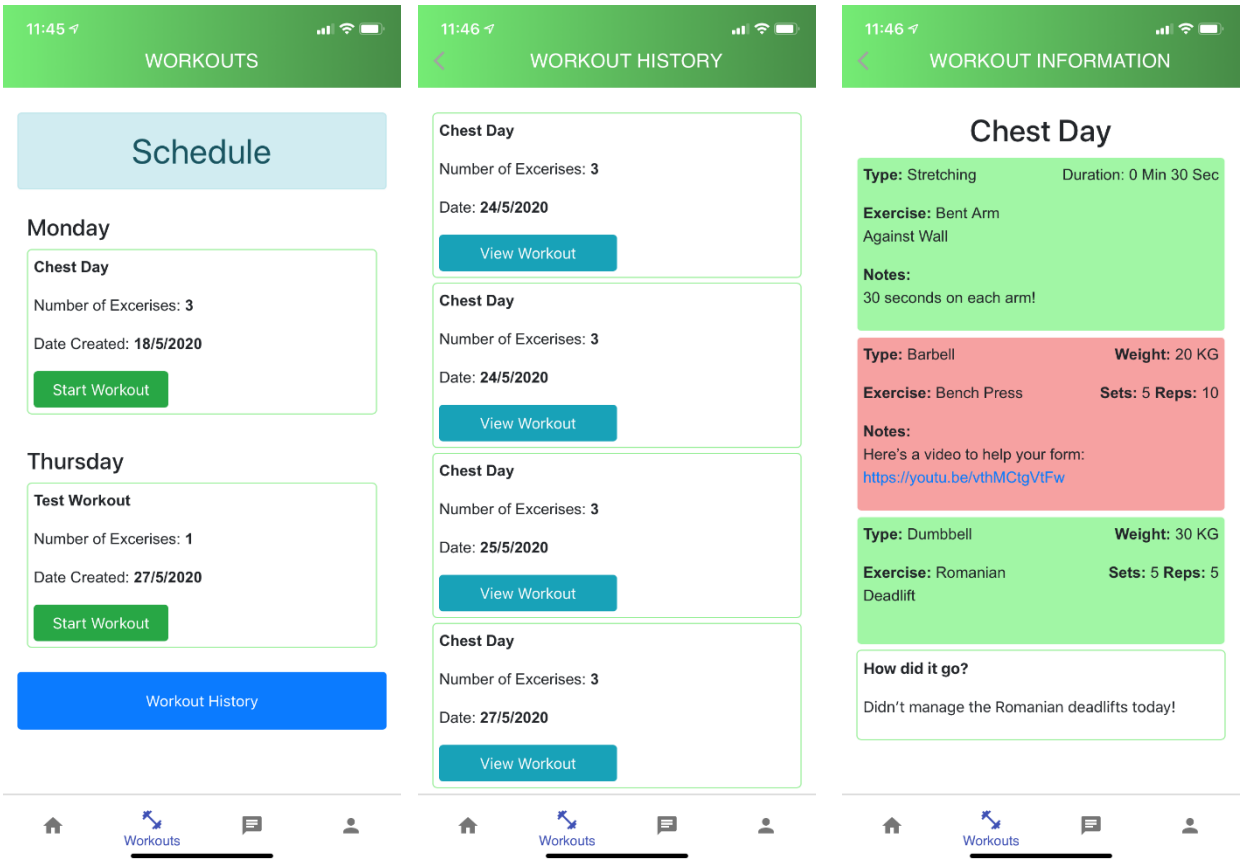






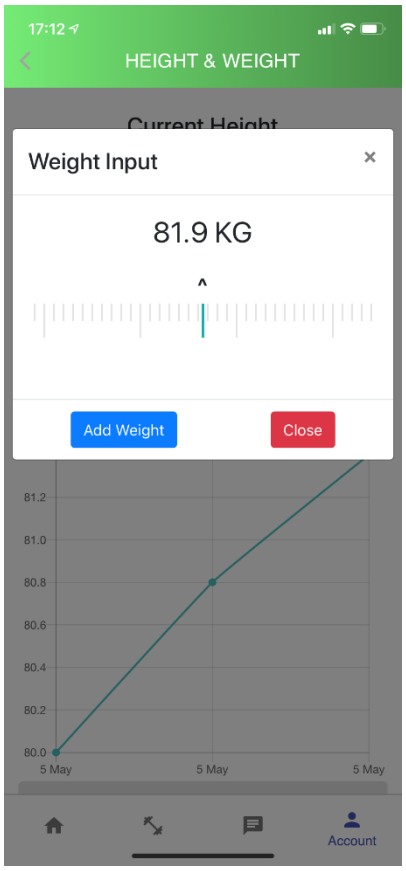
15.2.11 View Workout History

To view your workout history, first navigate to the workouts page of the application. From there click ‘Workout History’ at the bottom of the page. Then a list of all your previously recorded workouts will show, with the date that they were recorded. You can then click ‘View Workout’ to see a specific workout and what exercises were completed or missed.



15.2.12 Update Weight

To update your weight, first navigate to the account page of the application. From there click ‘Update Weight and Height’. This will display your current weight and height, along with a graph of all your previous weights. Click ‘Update Weight’ a slider will appear for you to be able to choose your current weight. Then click ‘Update’, your weight will be saved, and the graph will be refreshed to show the new weight.

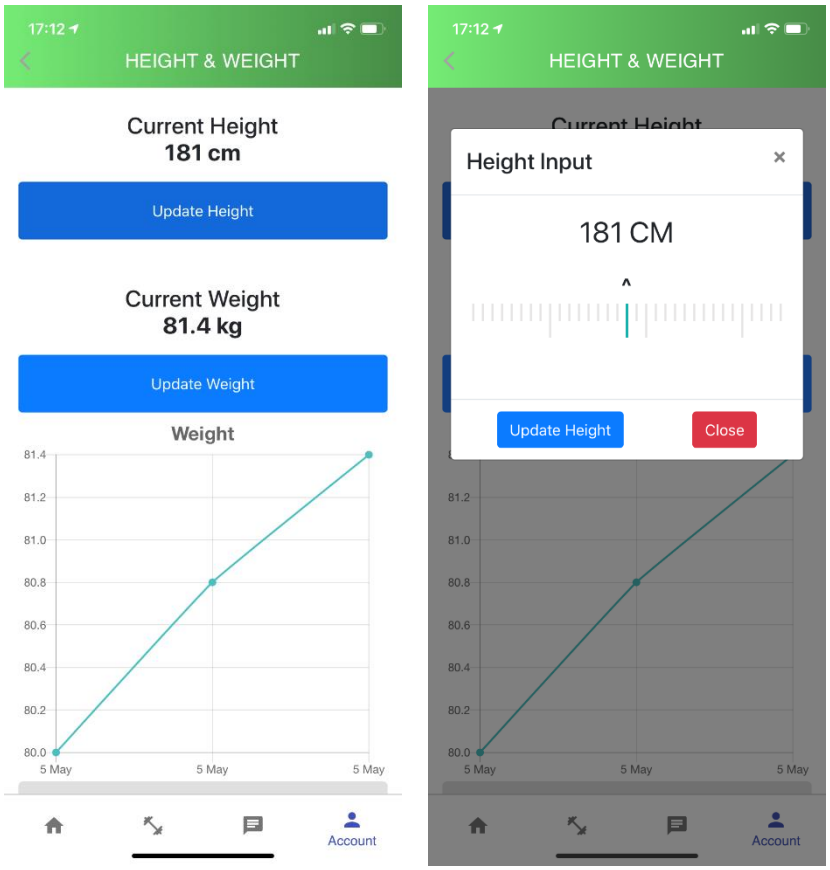


15.2.13 Delete Weight

To delete a previously recorded weight, navigate to the account page of the application. Then click ‘Update Weight and Height’, scroll to the bottom of the page to view a list of your recorded weights. A weight can be deleted by clicking the trash icon next to that weight. Once the weight it deleted the graph and list will refresh.

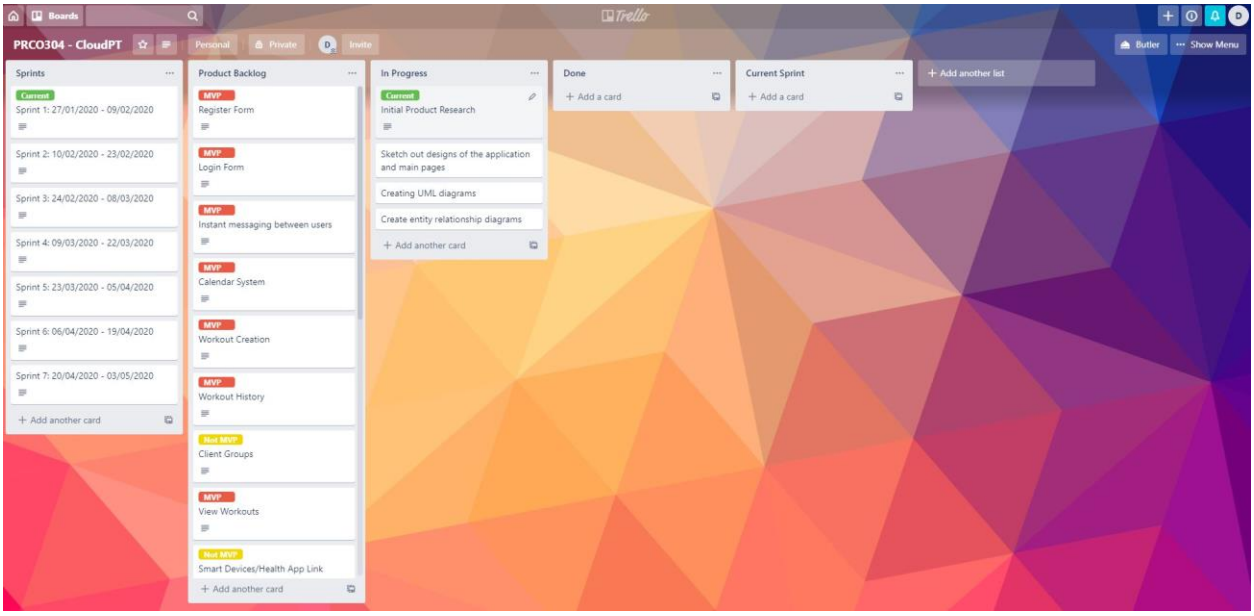
15.2.14 Update Height

To update your weight, first navigate to the account page of the application. From there click ‘Update Weight and Height’. Then click ‘Update Height’ and use the scroll bar to select your current height. Once the height has been updated the page will refresh to show your new current height.

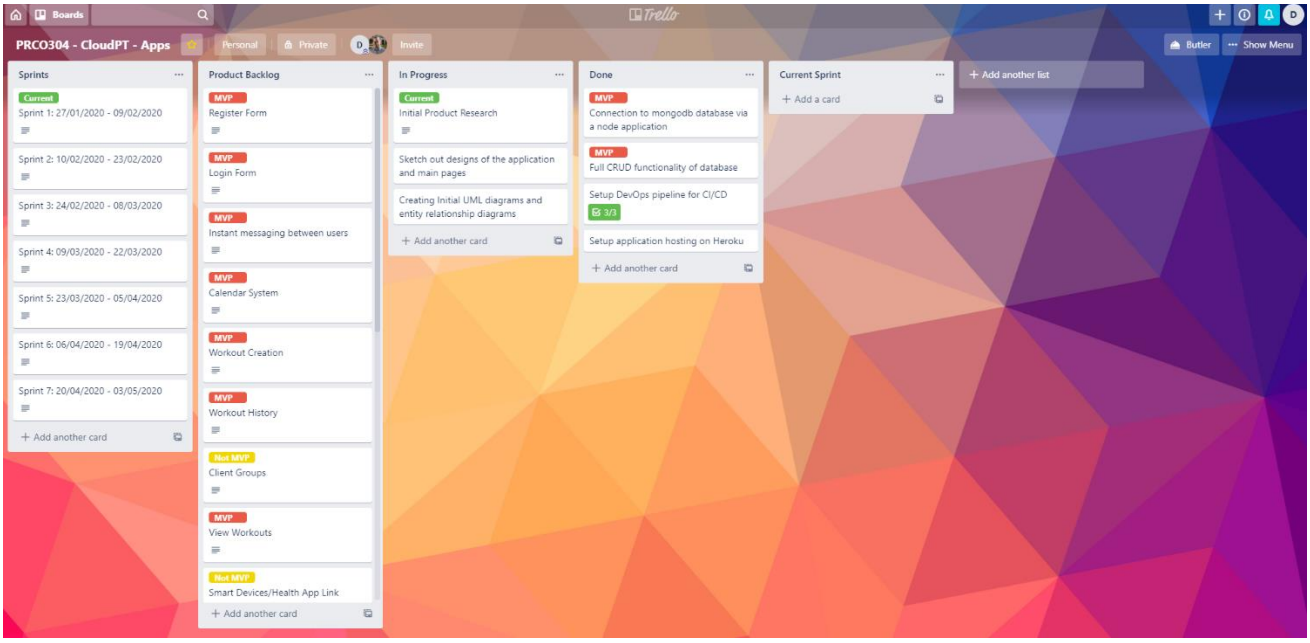


15.3 TRELLO BOARDS

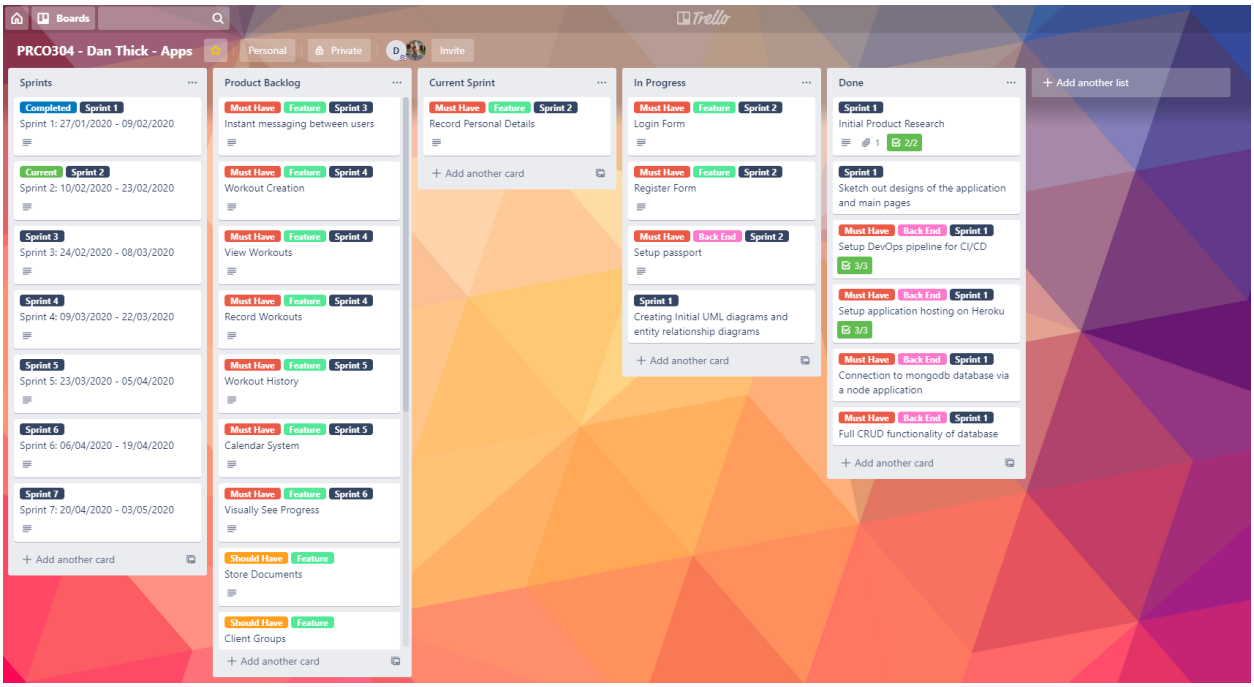
15.3.1 Board 1 (31/01/2020)



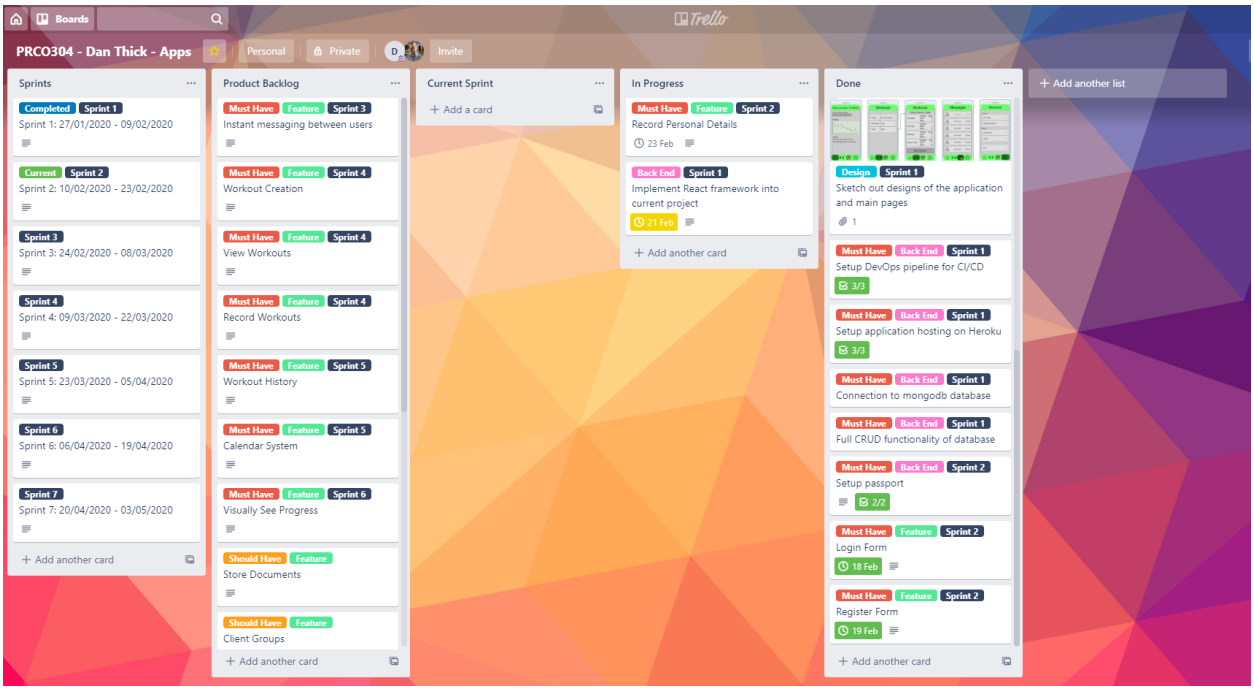
15.3.2 Board 2 (07/02/2020)



15.3.3 Board 3 (14/02/2020)

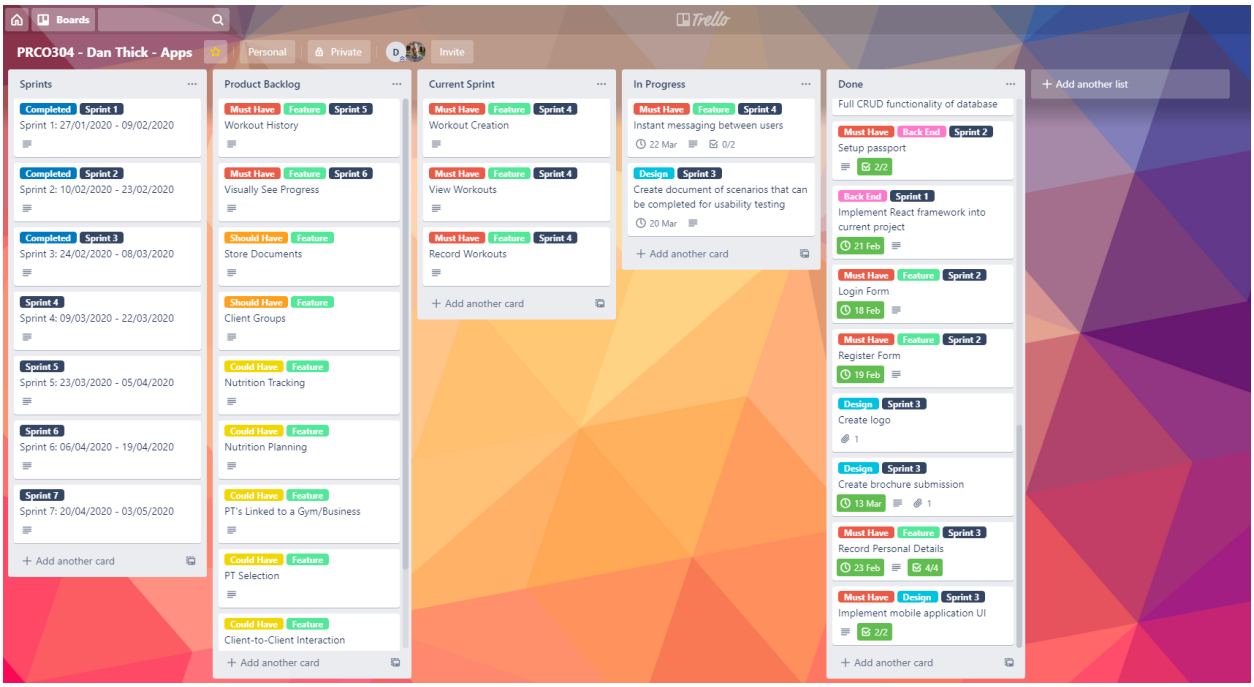


15.3.4 Board 4 (21/02/2020)

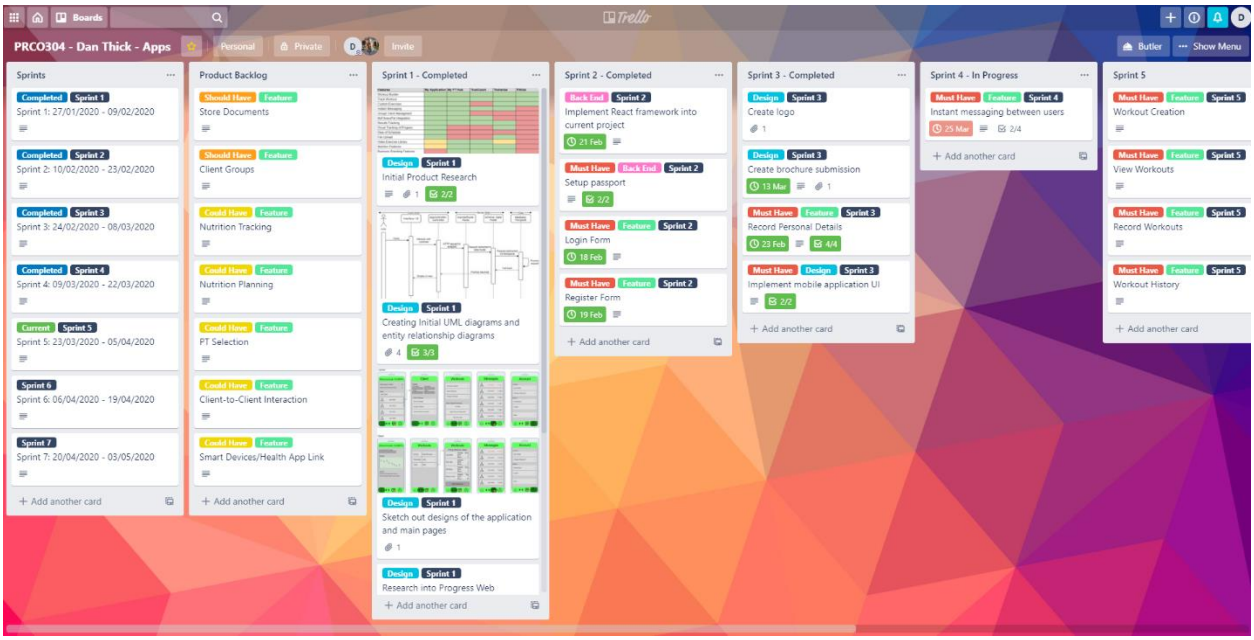


15.3.5 Board 5 (28/02/2020)

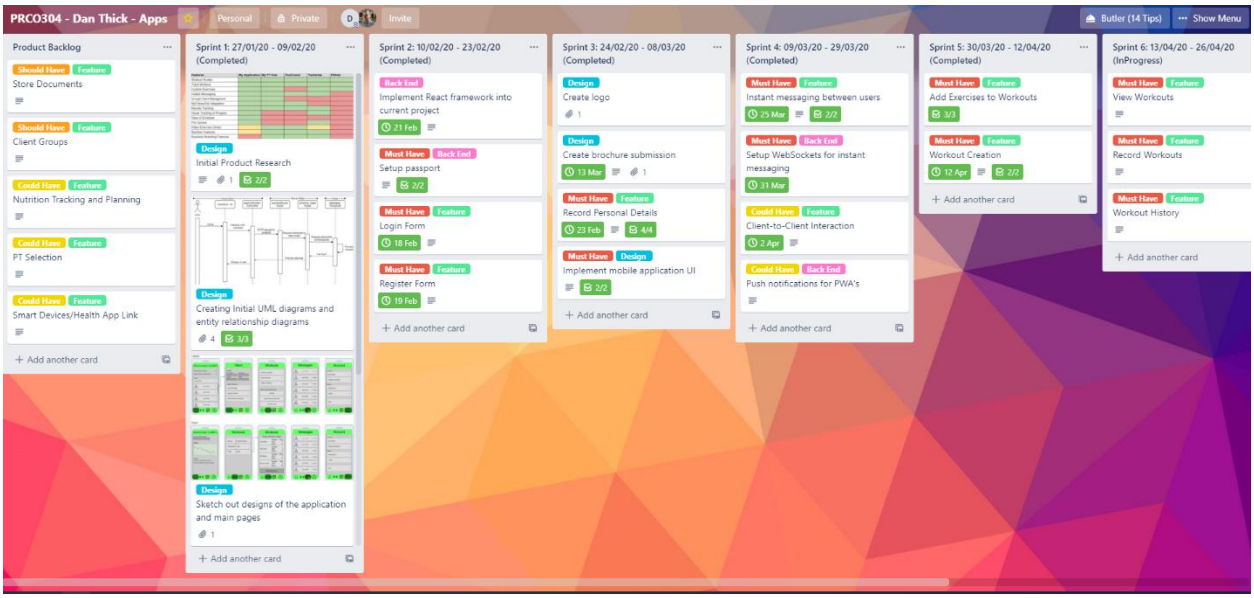
15.3.6 Board 6 (13/03/2020)



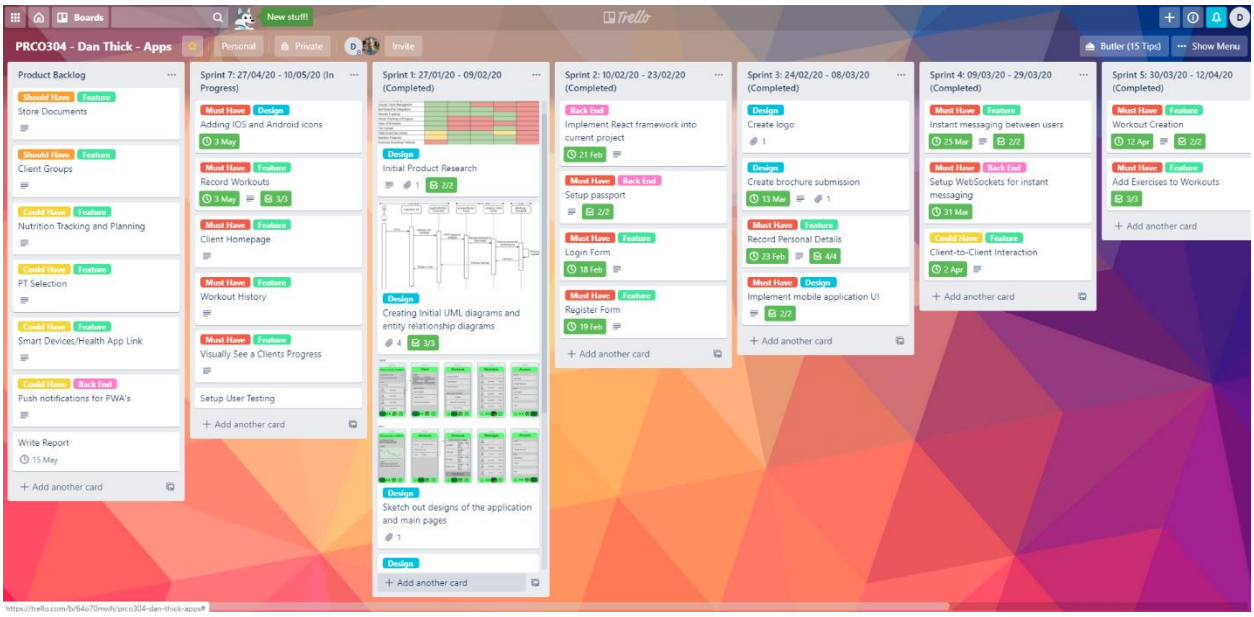
15.3.7 Board 7 (27/03/2020)



15.3.8 Board 8 (24/04/2020)



15.3.9 Board 9 (08/05/2020)



15.4 User Stories

User stories were obtained before creating the application's aims and objectives. The user stories were used and turned into functional requirements for the minimum viable product and for additional features of the application.

- As a user I would like to have a responsive application to use
- As a user I would like to be able to instant message other users on the application
- As a user I would like to be able to have a personal log in
- As a personal trainer I would like to be able to create a workout
- As a personal trainer I would like to be able to assign workouts to specific clients
- As a personal trainer I would like to be able to view client's past workouts and their progression
- As a personal trainer I would like to be able to group together clients to quickly assign workouts
- As a personal trainer I would like to be able to advertise to potential new clients on the application
- As a client I would like to be able to track my weight
- As a client I would like to be able to view workouts that have been assigned to me
- As a client I would like to be able to record a workout
- As a client I would like to be able to view previous workouts and how they went

- As a client I would like to be able to store files/documents that can be viewed by my personal trainer
- As a client I would like to be able to track my nutrition
- As a client I would like to be able to import data from a smart device

15.5 SMART OBJECTIVES

Below is a SMART break down for each of the minimum viable product objectives.

SMART Goal	**Design for mobile use and has a responsive and user-friendly user interface.**
Specific	The application needs to have a friendly-user interface that behaves like a native mobile application. There needs to be a bottom navigation bar, as well as a top app bar.
Measurables	This goal will be achieved when the overall feel of the application is very similar to a native mobile application.
Achievable	To achieve this goal, ReactJS and Bootstrap will be used to assist build the front end.
Relevant	This goal needs to be achieved to allow users to be able to use the application whilst they are at the gym. This means that they will have to use it on a mobile device.
Timely	This goal will be completed throughout the whole project. The initial UI implementation will be completed in the second sprint.

SMART Goal	**Allows for clients to track their weight.**
Specific	Clients need to be able to track their weight. All of their weights need to be stored and also be visible in a graph format.
Measurables	This goal will be achieved when there is functionality for weight input for a specific user and when a responsive graph is visible to the user.
Achievable	User weight data will be stored in its own table in the database and ChartJS will be used to produce a graph.
Relevant	This goal is needed to allow for clients and personal trainers to keep track of their weight and to view progress.
Timely	This goal should be achieved in sprint 3.

SMART Goal	**Allows for instant messaging between users.**
Specific	Users need to able to communicate with each other via instant messaging.

	Using email addresses, users should be able to start new chats with each other.
Measurables	This goal will be achieved when a user is able to start a new chat, view previous chats and send/receive messages with other users.
Achievable	Messages will be stored in the database and web sockets will be used to allow for real-time updates during conversations.
Relevant	This goal needs to be achieved to ensure that clients and personal trainers are able to stay in communication within the application.
Timely	This goal should be achieved in sprint 4.

SMART Goal	**Allows for personal trainers to be able to create workouts and assign them to their clients.**
Specific	Personal trainers need to be able to create custom workouts made up of exercises that can be added to the workout. They also need to be able to assign the workouts that have been created to their clients.
Measurables	This goal has been achieved once it is possible for personal trainers to save their custom workouts that they have created. As well as being able to assign these workouts to specific clients on specific days of the week.
Achievable	This goal will be achieved by allowing personal trainers to be able input all of the required details for each exercise. A workout will be made up of an array of exercises which will all be stored in the database.
Relevant	This goal is needed to allow for personal trainers to build programs for their clients, to allow them to follow.
Timely	This goal should be achieved in sprint 5.

SMART Goal	**Allows for clients to be able to view workouts assigned to them and log what they completed during a workout.**
Specific	Clients need to be able to see all of their workouts that are assigned to them and what day they need to be completed. Clients also need to be able to record what they achieved during the workout for their personal trainers to be able to see.
Measurables	This goal will be met when clients are able to view, in a schedule, a list of their assigned workouts. Also, when clients are able to record and complete a workout.
Achievable	This will be achieved by showing clients their assigned workouts with the ability to start a workout, with then the required options to record their workout.

Relevant	This goal must be achieved as it will allow both clients and personal trainers to see their progress over time.
Timely	This goal should be achieved in sprint 6.

SMART Goal	**Allows users to be able to see past workouts and their progress made.**
Specific	Both personal trainers and clients need to be able to view their past workouts to help them understand their progress.
Measurables	This goal will be achieved when clients are able to view a list of all their previous workouts and personal trainers are able to do the same for each of their clients.
Achievable	This goal can be achieved by displaying all of the required information in a way which all users can understand.
Relevant	This goal is necessary because it allows for tracking of progress of all clients.
Timely	This goal should be achieved in sprint 7.

15.6 EXISTING PRODUCTS

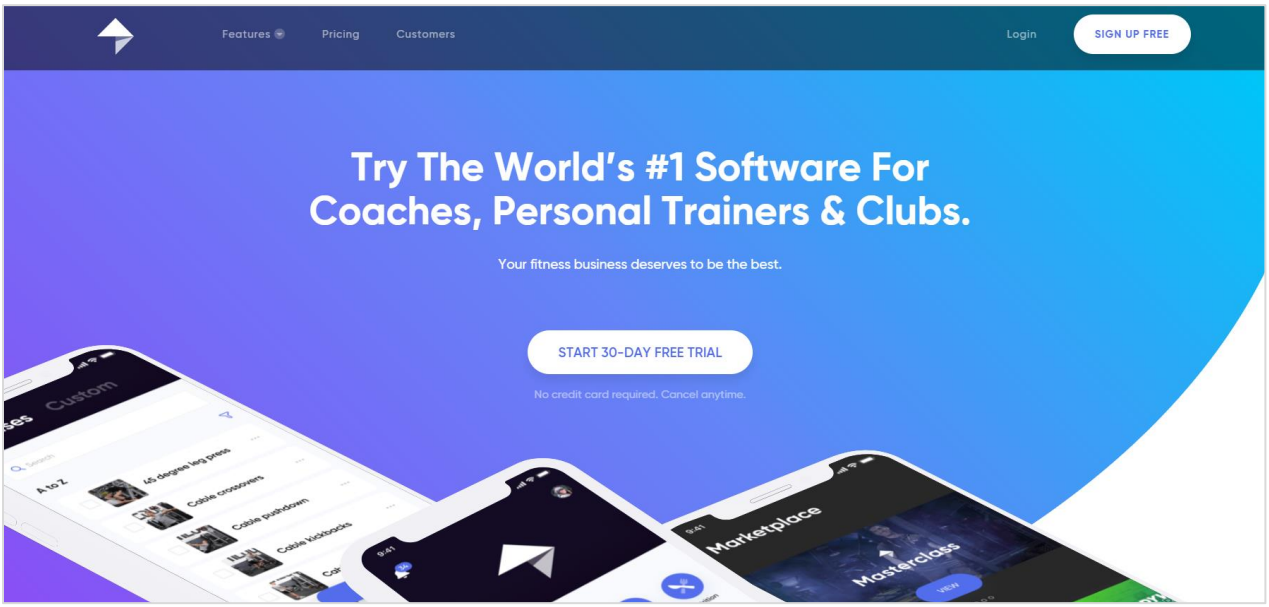
15.6.1 Comparison Table

Below is the comparison table that was created after completing some research as to what other products were on the market and were trying to solve a similar problem. The 'features' list consists of those that are planned to be in this application as well as features that were available in other products but are not planned for this application. Green shows the features that are the existing products currently have and what is planned for a minimum viable product. Red shows feature not included in the products. Yellow shows additional features that could potentially be included.

Features	CloudPT	My PT Hub	TrueCoach	Trainerize	Fithub
Workout Builder					
Track Workout					
Custom Exercises					
Instant Messaging					
Results Tracking					
Visual Tracking of Progress					
File Upload					
Video Exercise Library					

Nutrition Features					
Business Branding Features					

15.6.2 My PT Hub



My PT Hub is by far the most popular app used by personal trainers with over 85,000 signed up, along with 2,300,000 clients. They have had over 12,000,000 workouts created using the application, some of which are available on their marketplace (My PT Hub, 2020). They allow for unlimited workouts to be created which clients are able to see, to be able to track and log their workouts. Nutrition tracking is also available, allowing personal trainers to set a number of calories and macros to a client for them to be able to follow.

My PT Hub has made it very easy to manage a large number of clients, through grouping them together rather than working with each client one-on-one. It is possible to advertise your own personal brand by being able to completely customise the applications, logo, and themes for your specific clients.

What My PT Hub does lack is instant messaging to clients and the ability for clients to upload files for their personal trainers to see. In addition to this, a monthly charge is set for all users, with only 2 tiers of pricing, depending on the size of your business. Users are able to either pay for 5 clients or unlimited clients. This can be seen in Figure 7. This means that it is not necessarily suitable for personal trainers who do not have a large client base, which limits the target audience and in turn pushes personal trainers away from using such applications.

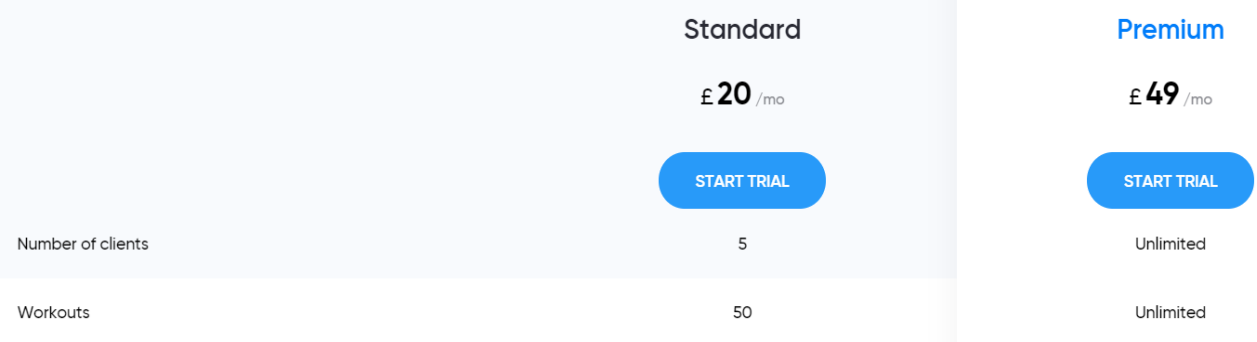
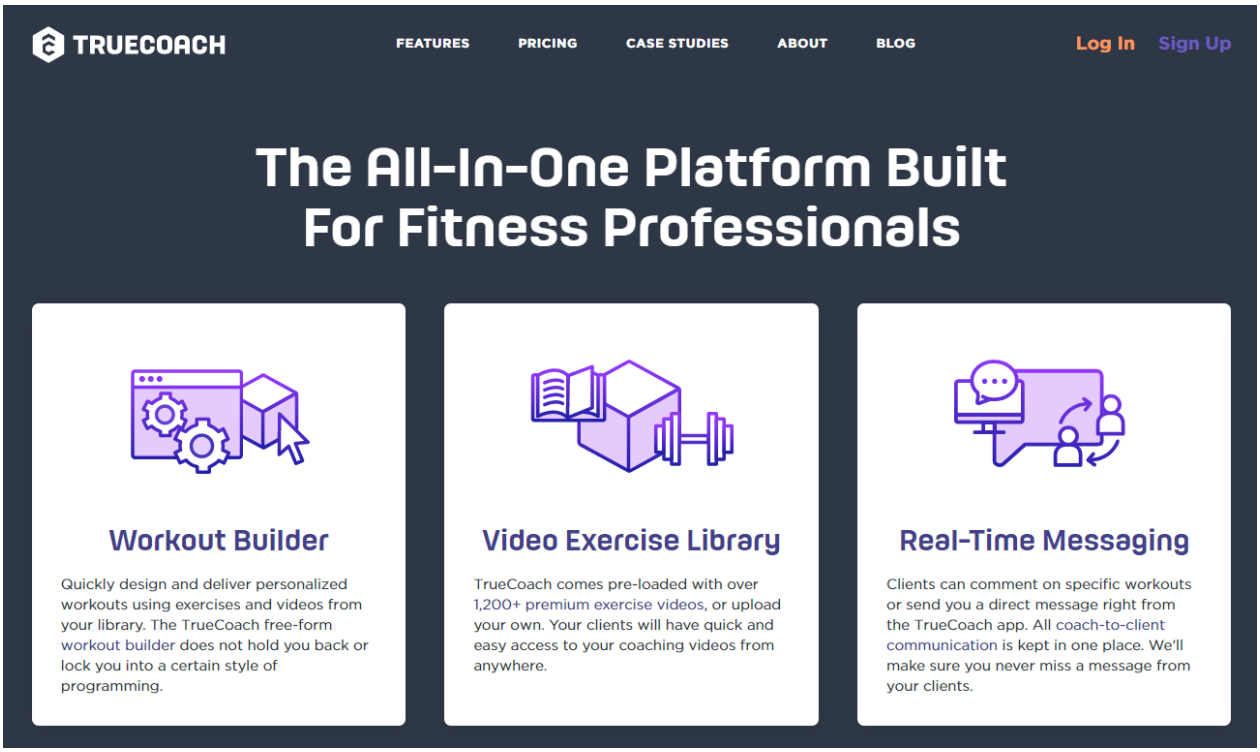


Figure 7. Pricing for My PT Hub (My PT Hub, 2020).

15.6.3 TrueCoach



TrueCoach is completely focused on building workouts and sharing them with clients. The application has over 17,000 personal trainers and 225,000 clients signed up with over 40,000,000 workouts delivered (TrueCoach, 2020). TrueCoach also includes a large library of videos for many exercises that can be included when creating workouts.

The problem with this application is that it does not allow for custom exercises to be added to workouts, meaning that the exercises you can choose from are fairly limited. In addition to this, personal trainers are restricted to the number of clients they can have, with the pro plan only allowing 50 clients for a large monthly fee (Figure 8).

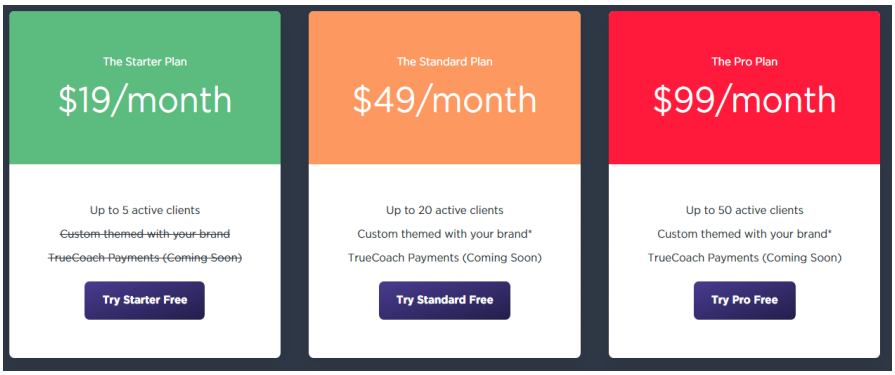
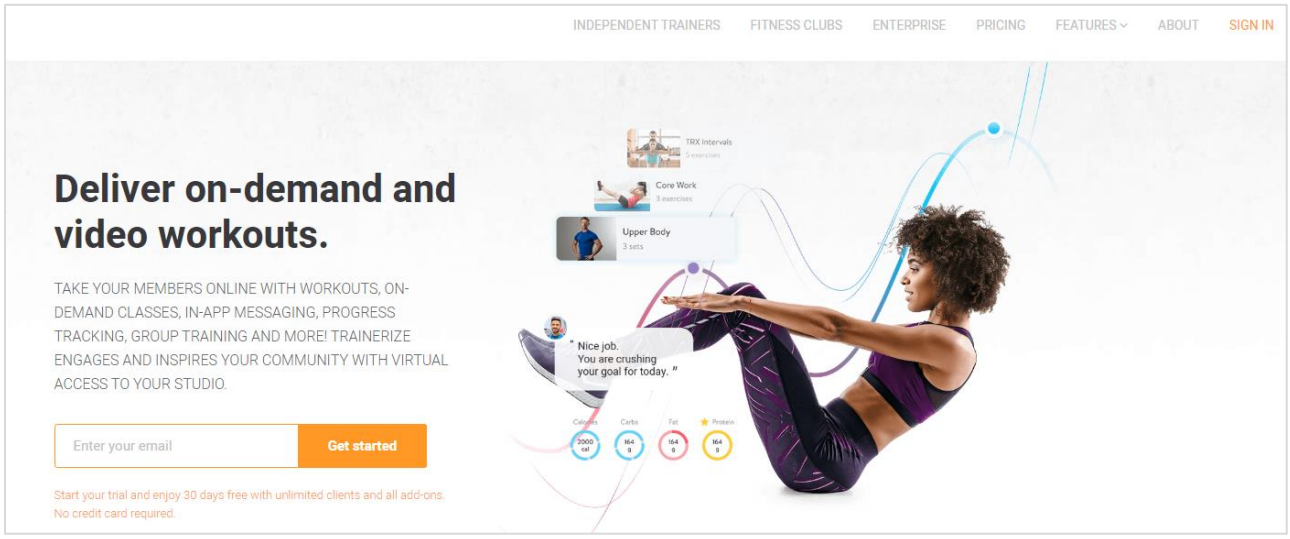


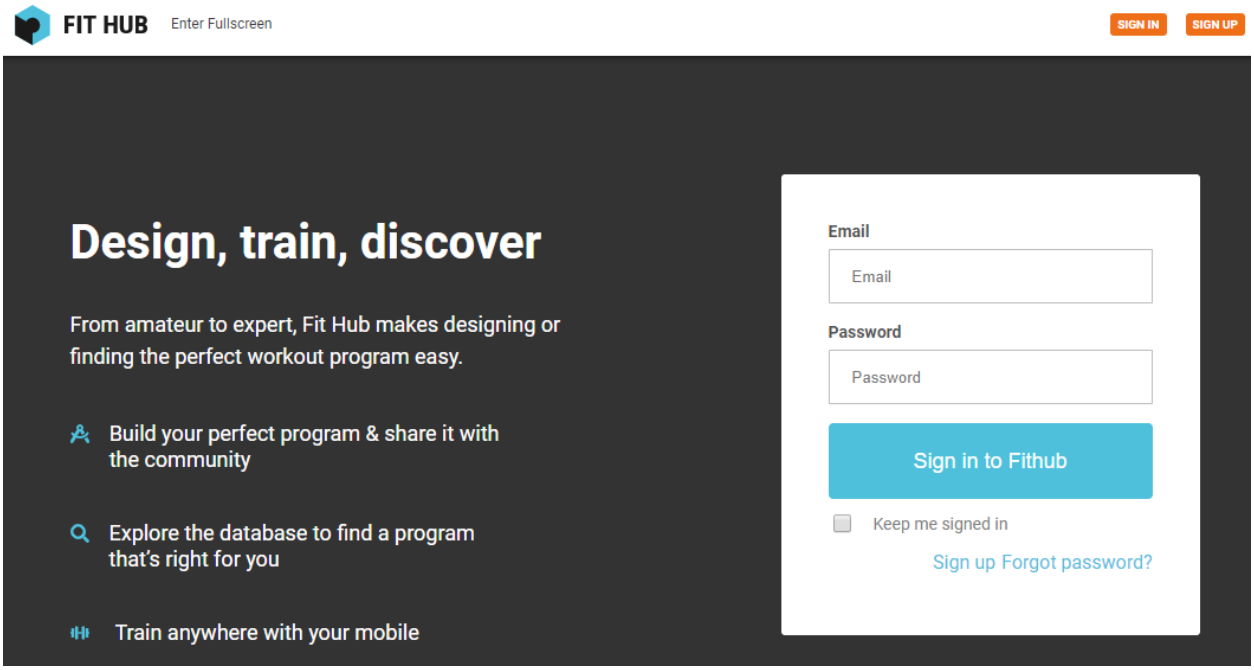
Figure 8. Pricing for TrueCoach (TrueCoach, 2020).

15.6.4 Trainerize



Trainerize allows for not only individual trainers to use the application, but also clubs and fitness businesses. This means that it has many more features aimed towards businesses using the application to interact with their clients, such as: payments, marketing material and a social platform. Trainerize is slightly different than what is planned for the new solution because of the fact that it is aimed more towards clubs and businesses.

15.6.5 Fithub

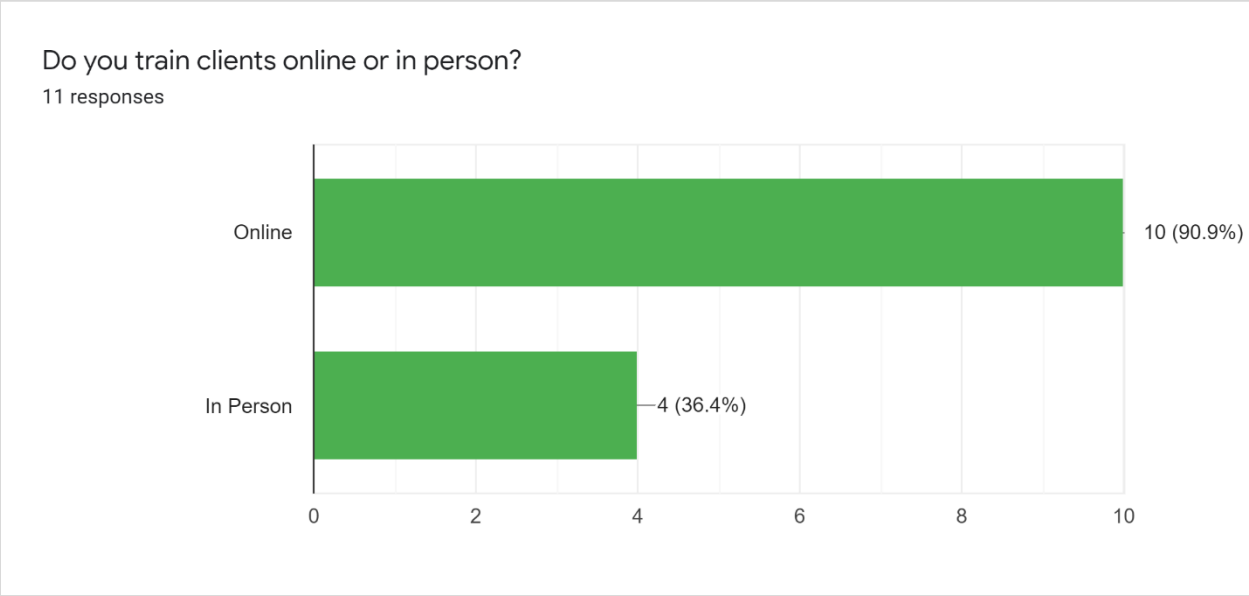


Fithub has the least features out of all the applications that were tested during the initial research. It allows for users to create workouts and complete workouts. However, it does not allow for personal trainers and clients to interact with each other. Personal trainers are not able to assign workouts to a specific client or are they able to communicate in anyway via the application. Creating a workout only allows for very basic options and cannot even be done whilst on a mobile device. However, the application is completely free to use, with the ability to create and complete an unlimited number of workouts.

15.7 Market Research Questionnaire

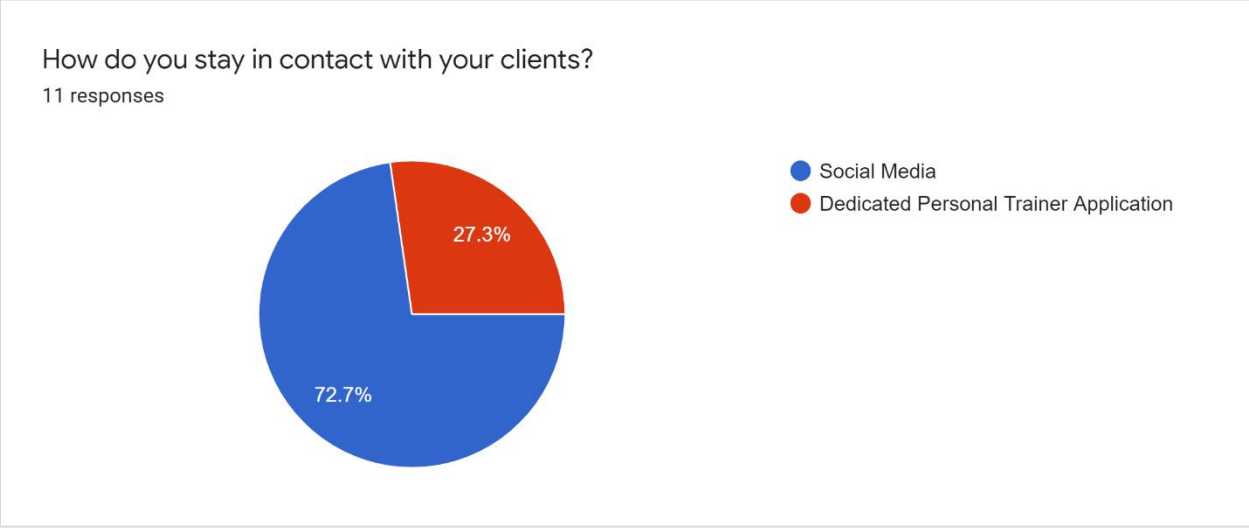
1. Do you train clients online or in person?
2. How do you stay in contact with your clients?
3. What application do you use to stay in contact with your clients?
4. Do you think that it is worth the cost?
5. What social media platform do you use to stay in contact with your clients?
6. How do you give workout programs to clients?
7. Do you find it easy to track a client's progress?
8. Have you ever considered using a dedicated personal training application?
9. Why are you note using a dedicated application?
10. If the right application came along (at the right price) would you consider using it?

Question 1



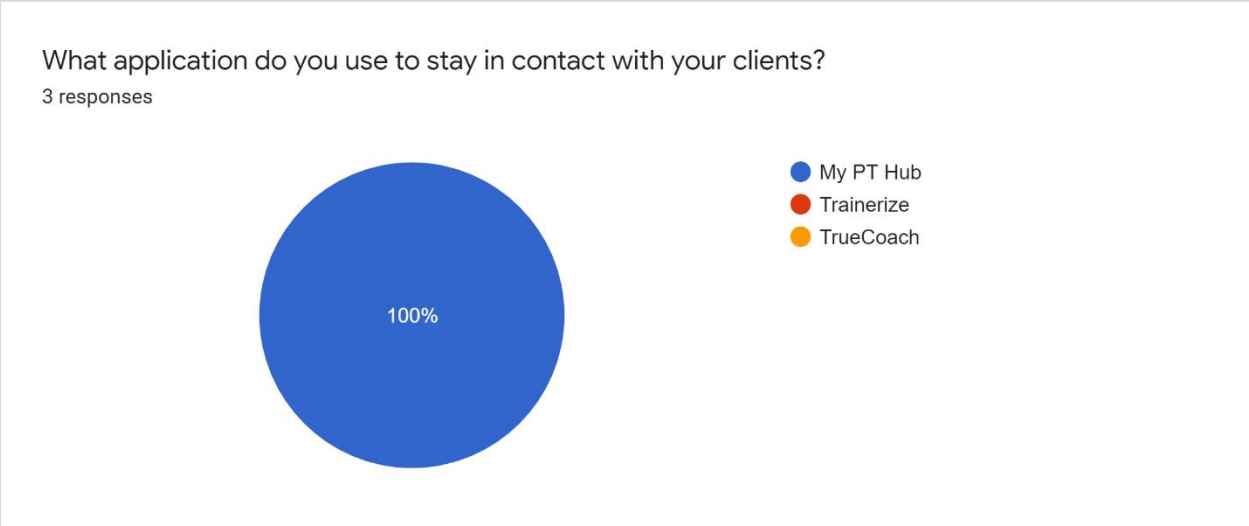
This question shows that 90% of respondents do use some form on online training for their clients. This goes to show just how popular online personal training has become, compared to only 36% of personal trainers training their clients face to face. This does also mean that some personal trainers are using a combination of both online and face-to-face training.

Question 2



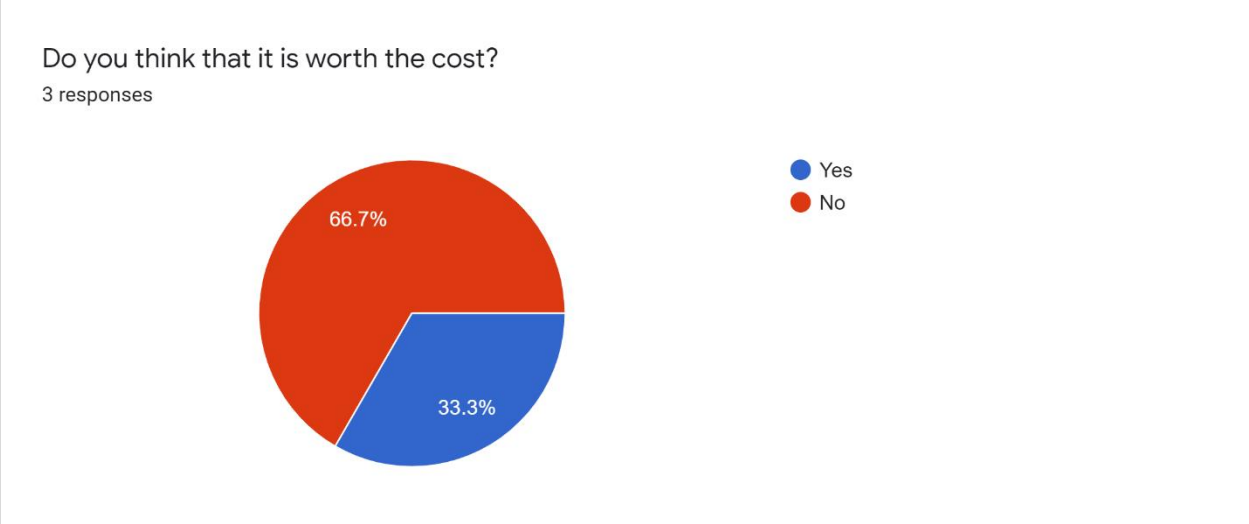
Over 70% of respondents only use social media to stay in contact with their clients instead of a dedicated personal training application. This shows that the marked uptake of these applications is relatively small for the target audience.

Question 3



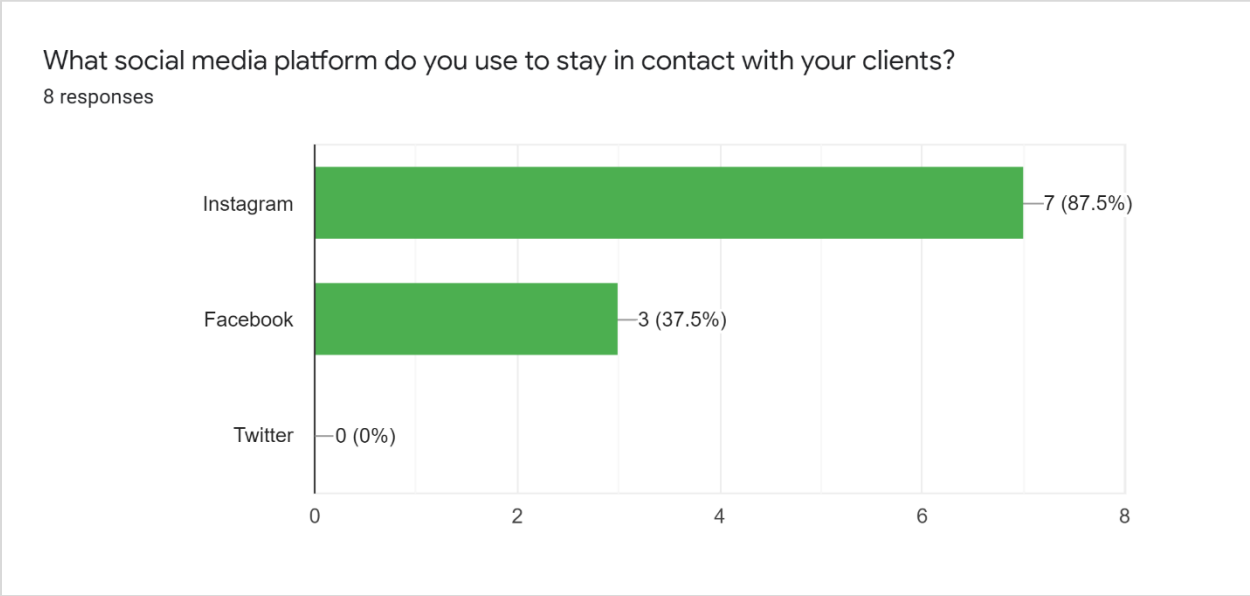
This question was only answered by respondents who use a dedicated personal training application. All of which use My PT Hub. This makes sense as in the UK it is the most popular application, whereas other competitors are more popular in other countries.

Question 4



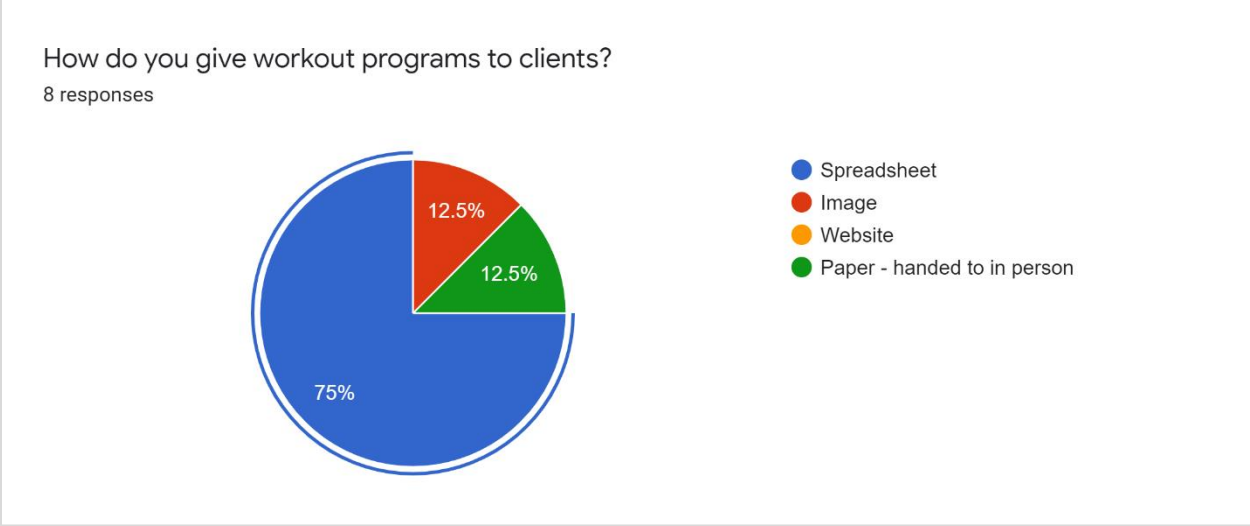
Again, this question was for respondents who use a dedicated application. Only 1 of the respondents thought that the cost of the application was worth it to them.

Question 5



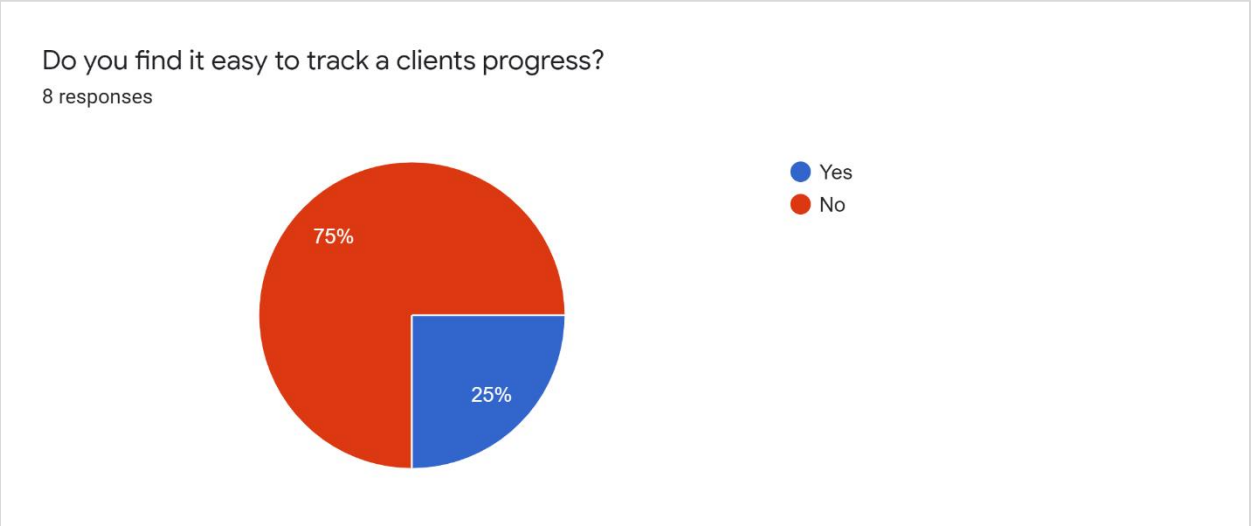
Respondents who only use social media were then asked what platforms they use, with Facebook and Instagram being the most popular.

Question 6



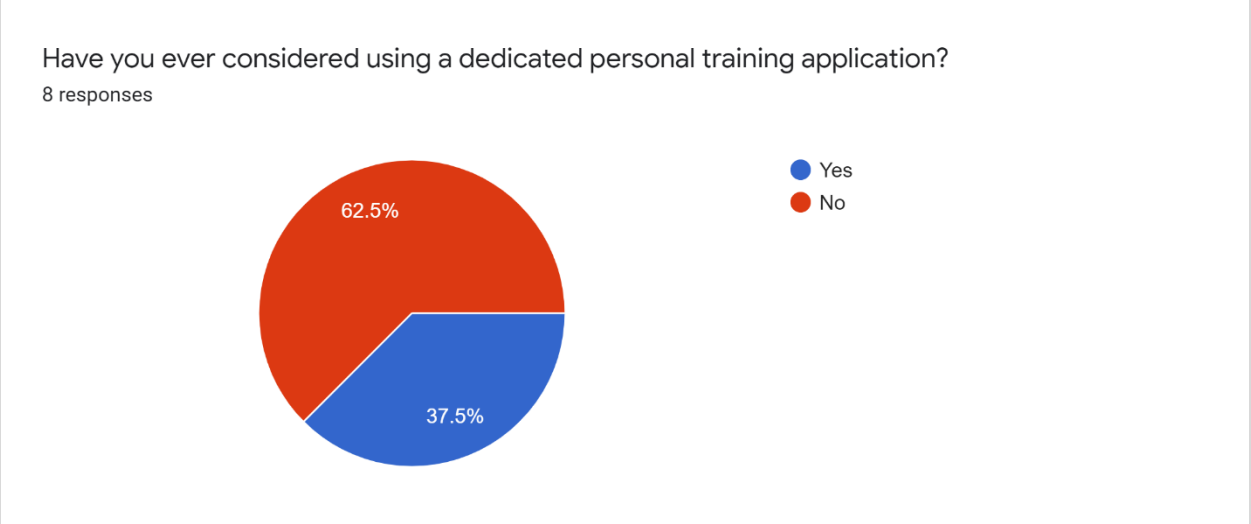
The majority of personal trainers who use social media also use a spreadsheet to give clients workout programs.

Question 7



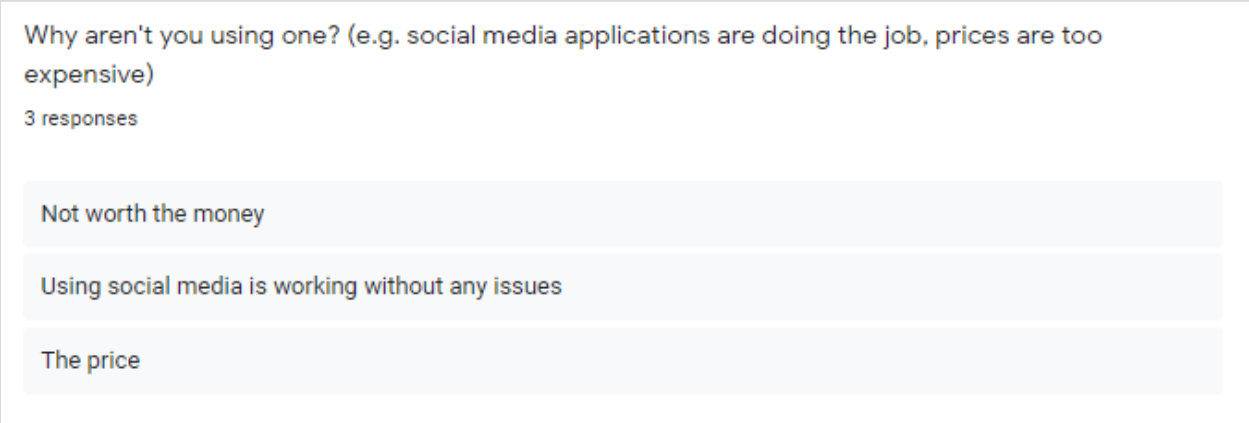
75% of trainers who only use social media, do not find it easy to track their client's progress. This shows that there is a problem that does not solving for these users.

Question 8



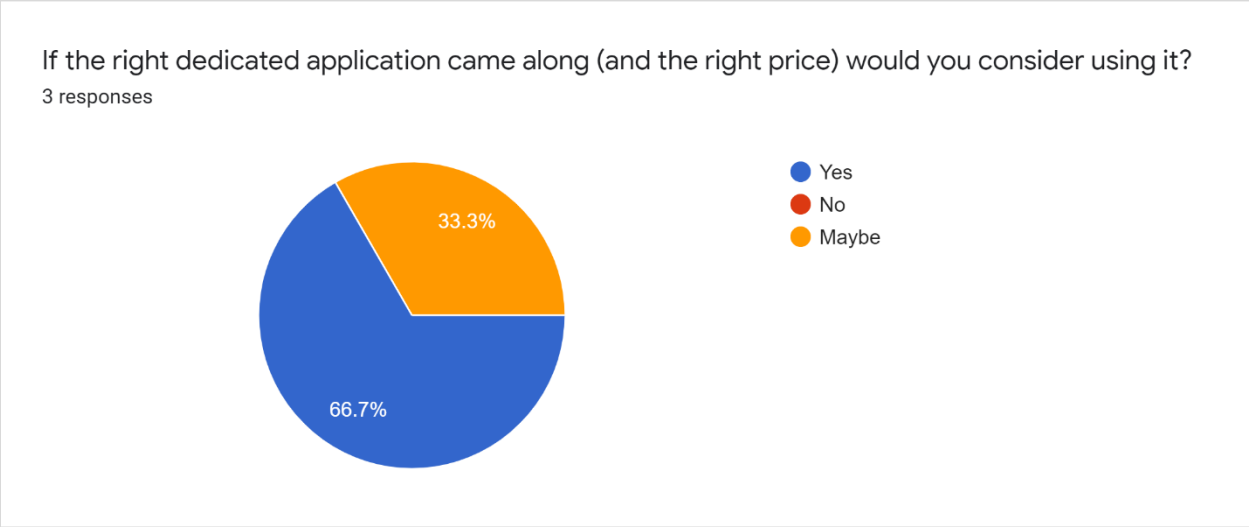
This question shows that many personal trainers have not considered using a dedicated personal training application. This could perhaps be because other platforms are working well for them or they do not think it would be worth the price.

Question 9



This shows that people have not considered using a specific application due to the cost and because social media works well for them.

Question 10



100% of responders would potentially, if not definitely, consider using a dedicated personal training application instead of just a social media platform. This is considering if the application has the correct features and is not too expensive.

15.8 Project Showcase Brochure

Project Title
CloudPT – Personal Training Management System

Student Name
Daniel Thick

Course Title
BSc (Hons) Computer Science

Abstract Text

CloudPT is a progressive web application designed to improve communication between personal trainers and their online clients.
Using a client management system, personal trainers can view clients progress and workout history. They can create and assign workouts to specific clients and instant message them. Clients can view assigned workouts and input details about completed workouts. They can track their weight and view their progress across exercises as well as instant message their personal trainer.
CloudPT has been developed with JavaScript using the MERN stack. Web sockets have been used to allow for communication between clients and it is being deployed via Heroku.

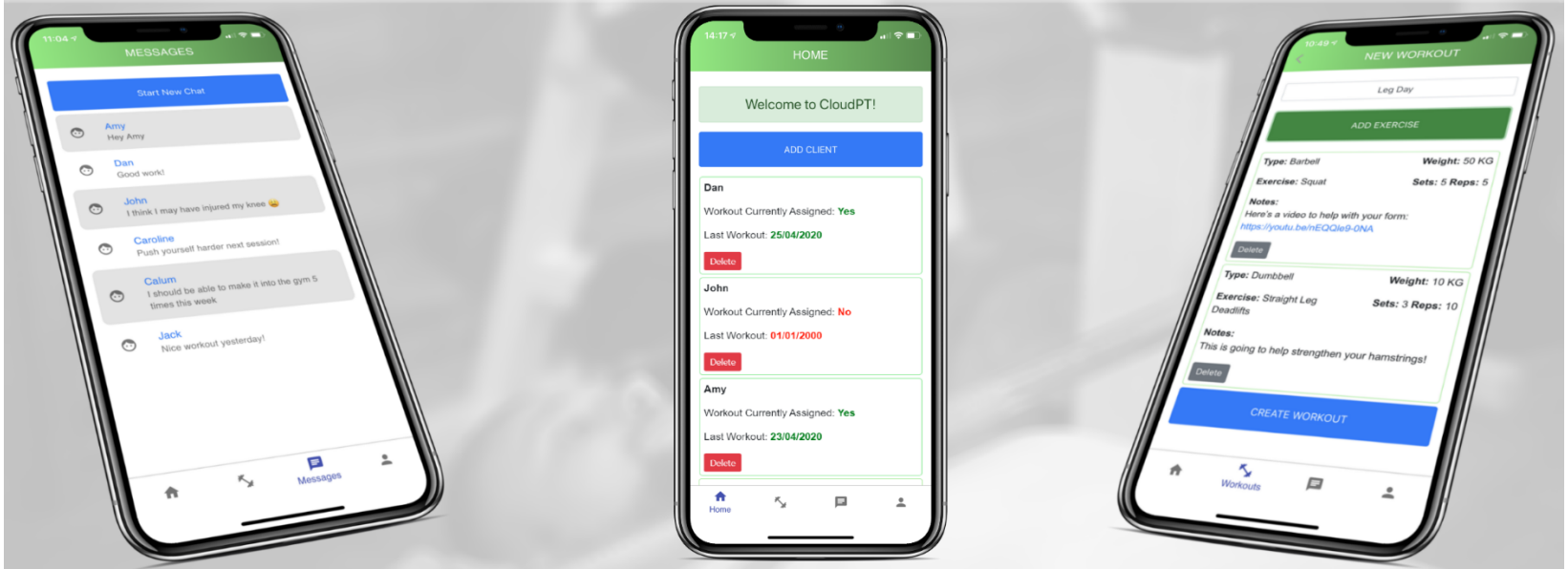
Technologies Used

- MongoDB
- Express
- ReactJS
- Node.js
- Heroku

Logo

CloudPT

Take Personal Training to the Cloud



Features

- Manage Clients
- Custom Workout Builder
- View Clients Workout History
- Instant Messaging
- Track Progress

Technologies

- Built using the MERN stack.
- Designed as a progressive web application to allow for cross-platform support.
- Hosted on Heroku.

Daniel Thick
daniel.thick@students.plymouth.ac.uk
cloudpt.me

BSc (Hons) Computer Science
Final Year Project

UNIVERSITY OF PLYMOUTH

15.10 PRODUCT BACKLOG

Objective	Sprint	MVP
Setup DevOps pipeline for CI/CD	1	Must Have
Setup application hosting on Heroku	1	Must Have
Setup connection to a mongodb database	1	Must Have
CRUD functionality of mongodb database	1	Must Have
Setup passport authentication	2	Must Have
Ability to login on homepage	2	Must Have
Ability to register a new account	2	Must Have
Ability to record personal details (name, weight, height)	3	Must Have
Implement mobile UI design	3	Must Have
Ability to instant message other users	4	Must Have
Implement WebSockets to allow for real-time updates	4	Must Have
Ability to create a new workout	5	Must Have
Ability to add exercises to a workout	5	Must Have
Ability to assign workouts to clients	5	Must Have
Ability to add new clients	6	Must Have
Ability to view a list of the users' clients	6	Must Have
Ability for clients to view their workout schedule	6	Must Have
Ability for clients to record their workouts	7	Must Have
Ability for both personal trainers and clients to see their workout history	7	Must Have
Ability to store documents	N/A	Should Have
Ability to group together clients	N/A	Should Have
Ability to track nutrition	N/A	Could Have
Ability for clients to choose a personal trainer from a list	N/A	Could Have
Ability to link smart devices	N/A	Could Have
Implement push notifications	N/A	Could Have

15.11 INITIAL PLAN

15.11.1 Project Plan

Stage	Start Date	End Date	Outcomes
Initiation	-	19/01/20	First proposal put forward and risk assessment created.
Market Research and Requirements	20/01/20	26/01/20	Conduct some market research about what solutions are currently used. List initial requirements and potential technologies to be used.
Sprint 1	27/01/20	09/02/20	Create high-level UML diagrams to show planned architecture design. Setup DevOps pipeline and hosting.

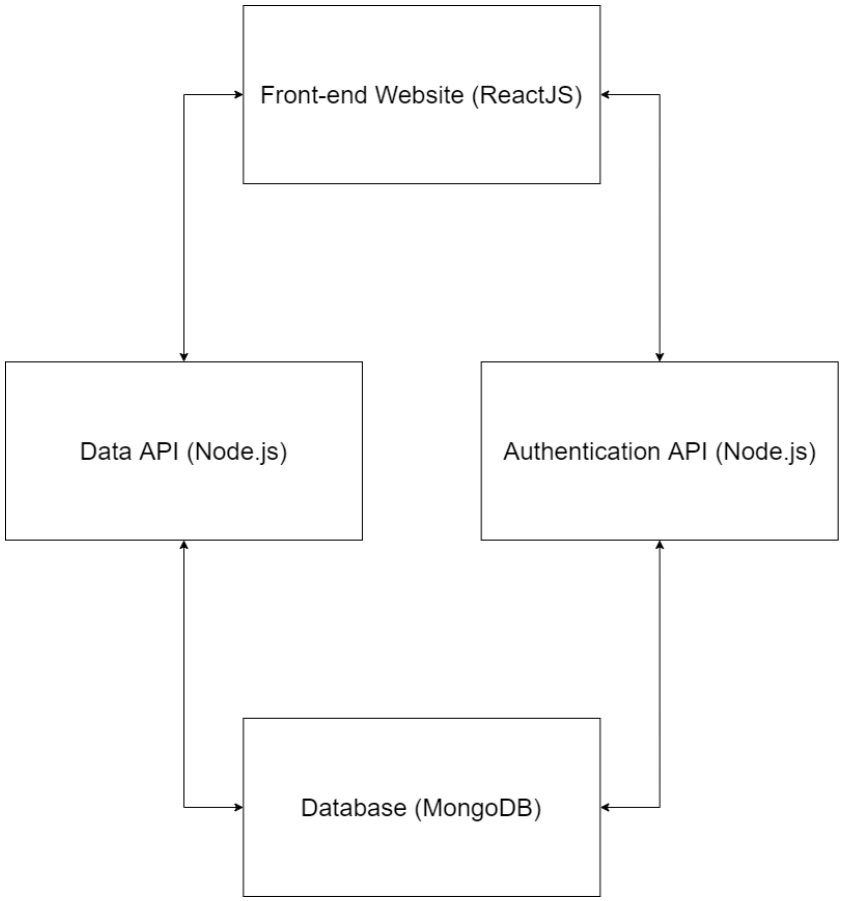
			Setup API with connection to a database.
Sprint 2	10/02/20	23/02/20	Implement authentication system with login and new user register functionality.
Sprint 3	24/02/20	08/03/20	Setup mobile UI with needed navigation components.
Sprint 4	09/03/20	29/03/20	Implement instant messaging between users using WebSockets.
Sprint 5	30/03/20	12/04/20	Begin implementation of personal trainer features for creating and assigning workouts.
Sprint 6	13/04/20	26/04/20	Implement client features for recording workouts and seeing past workouts.
Sprint 7 and Testing	27/04/20	10/05/20	Complete personal trainer features for viewing client's workouts and progress. Conduct user testing on finished product and make necessary changes.
Report Completion	11/05/20	24/05/20	Completed project report and proofread.
Submission Preparation	25/05/20	27/05/20	Create project video and prepare for viva with supervisor.

15.11.2 Risk Assessment

Impact	Risk	Strategy
High	Schedule overrun	Sprints will be used to ensure that the project stays on track throughout the entire duration. Sprint reports will take place at the end of each sprint to discuss what deliverables were met, what went well, and what did not go to plan. This will help adjust the following sprints if needed to allow for implementation that could not be completed in the previous sprint.
High	Overestimating technical ability	Being realistic with what is possible in each time frame is important. A feasibility study will be used to ensure that all of the objectives are possible to meet with the chosen technology. If a new technology must be learnt, then time will be allocated to ensure that this is possible.
Medium	Over scoping Project	An appropriately sized project will be chosen, with a feasible minimum viable product. This will allow for additional features to be added on top of the minimum viable product if the time

		restrictions allow for it. Using an agile approach allows for plans to be changed throughout the project.
Medium	Data loss	The use of version control will mean that the project will be backed up when any changes are pushed via Git. This will mean that if data loss occurs, it will be very minimal.
Low	Productivity setbacks	Keeping on track with the project plan is essential and will allow for goals to be met at specific dates. This will help ensure that all objectives are met, and the productivity levels stay high.
Low	Illness	Using an agile approach means that alterations can be made to the plan without having a significant impact on the project.

15.11.3 Architecture Design



15.12 GitHub

GitHub was used as the hosting platform for version control throughout the project. Figure 9 shows commits made to my final year project and a more up-to-date version can be found at: https://github.com/danthick/PRCO304.

Only one branch was used during the project, this is because there was only one developer working on it and the application was not in production during development so changes could be reverted easily. Each commit made to the repository contained a comment about what changes have been made and how the functionality of the application has changed. The repository also contains a read me file which describes the application, the installation process and states my supervisor.

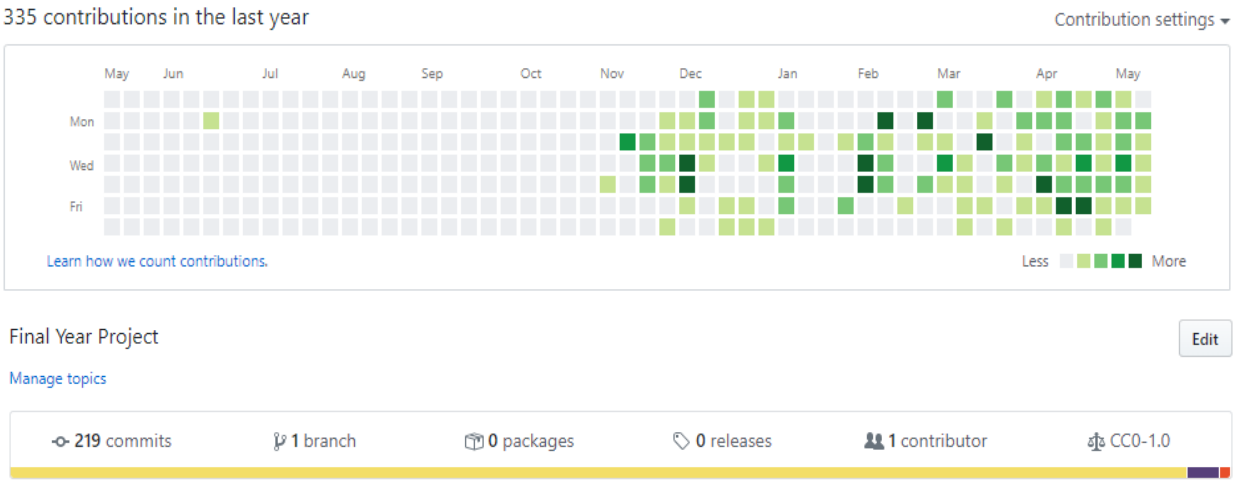


Figure 9. Commits made to Final Year Project repository.

15.13 Continuous Integration and Continuous Deployment

15.13.1 Travis CI

To ensure that all changes that were pushed passed all the tests, Travis CI was used to test all commits made to the projects GitHub repository. Figure 10 shows an example of the master branch passing a build. Once a build passes, it is then pushed to Heroku to be deployed. If the build fails, then the commit is not pushed to Heroku and an alert is emailed to the developer.

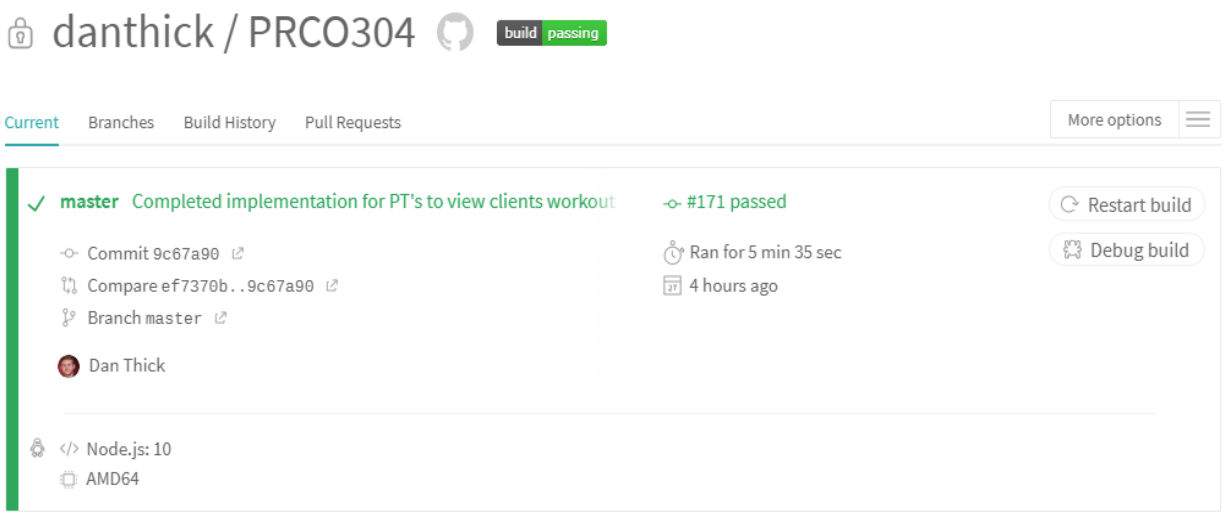


Figure 10. Travis CI successful build example.

A build status badge has also been added to the README.md of the GitHub repository. This allows visitors of the repository to see the current build status and clicking it will link to the Travis CI pages for that repository. This can be seen in Figure 11.

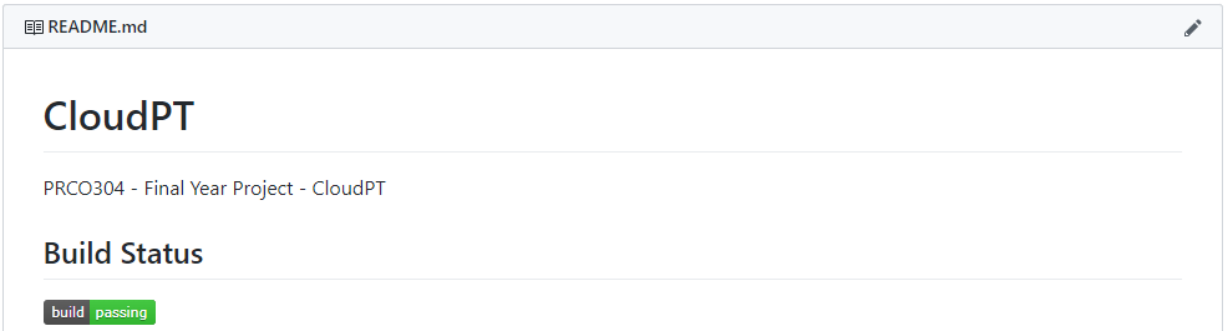


Figure 11. Repository README.md file containing a build status badge.

15.13.2 Heroku

Heroku was used to host the web application. A .travis.yml file was used in the project repository to setup deployment from Travis CI to Heroku. As shown in Figure 12, automated builds are created on Heroku and then deployed onto the live server. For more information about how the application is hosted, please see (Appendix 15.18).

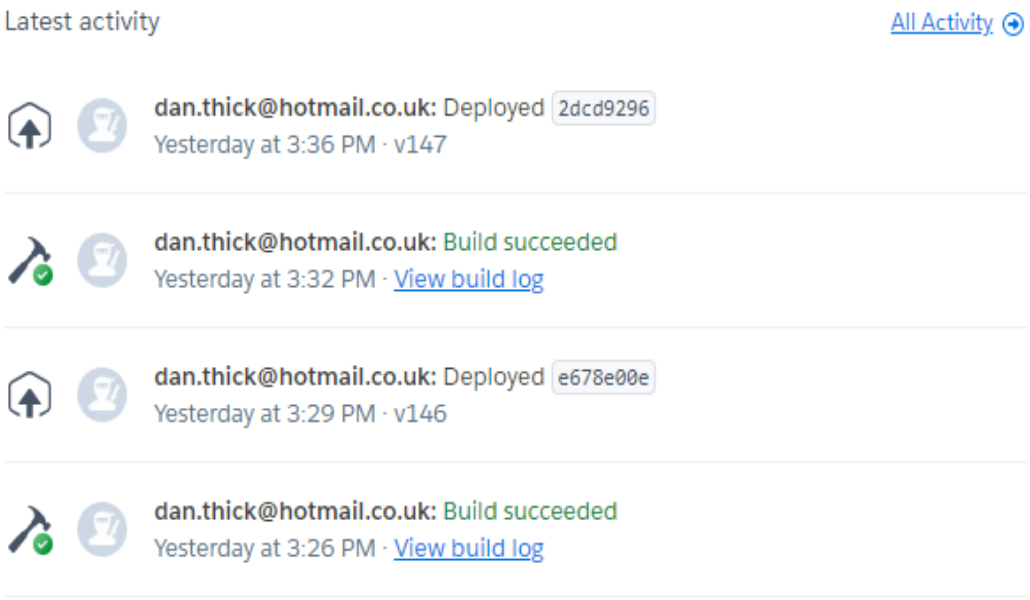
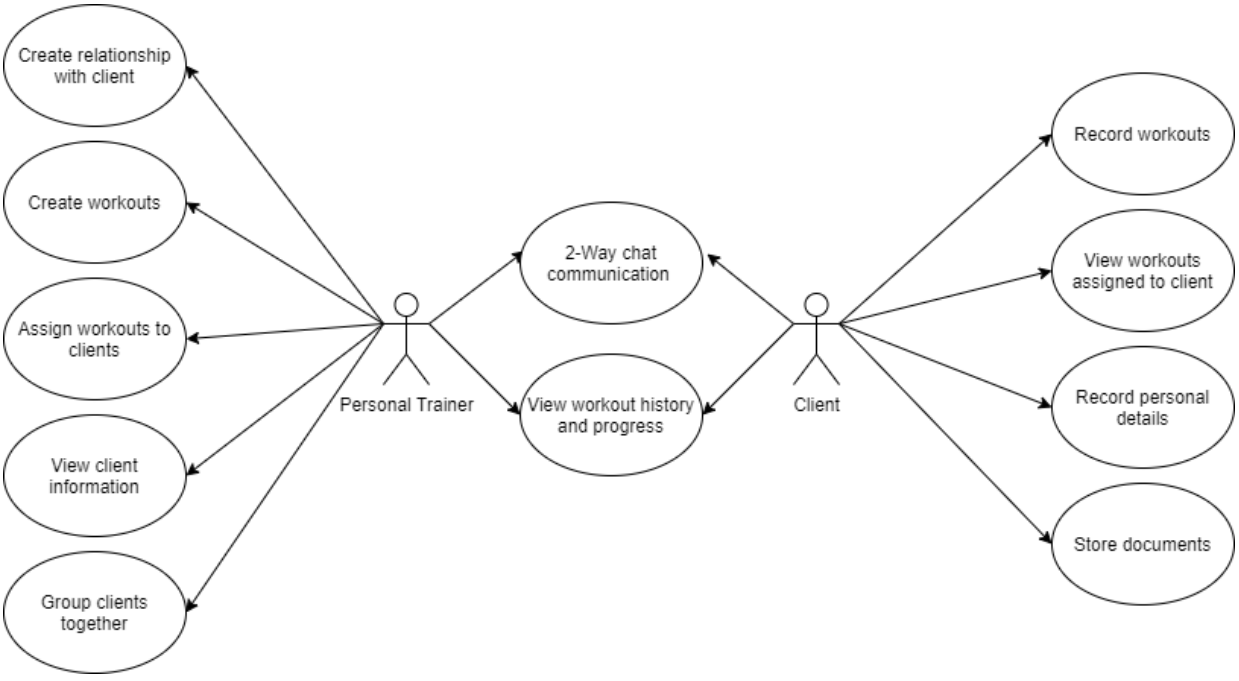


Figure 12. Automated deployment activity on Heroku.

15.14 USE CASE DIAGRAM

A use case diagram was produced at the start of the project as it is a good way to visualise how the users are going to interact with the system's functionality. It is a high-level view of the application's features and is simple to understand without showing how the system is designed.



15.15 SEQUENCE DIAGRAM

A sequence diagram was created to show how user interactions take place across the entire system. Sequence diagrams are timed focus and show in stages how actions interact with each part of the system and how data is returned. This sequence diagram, shown in Figure 13, describes what happens when a user clicks within the application to get or post data

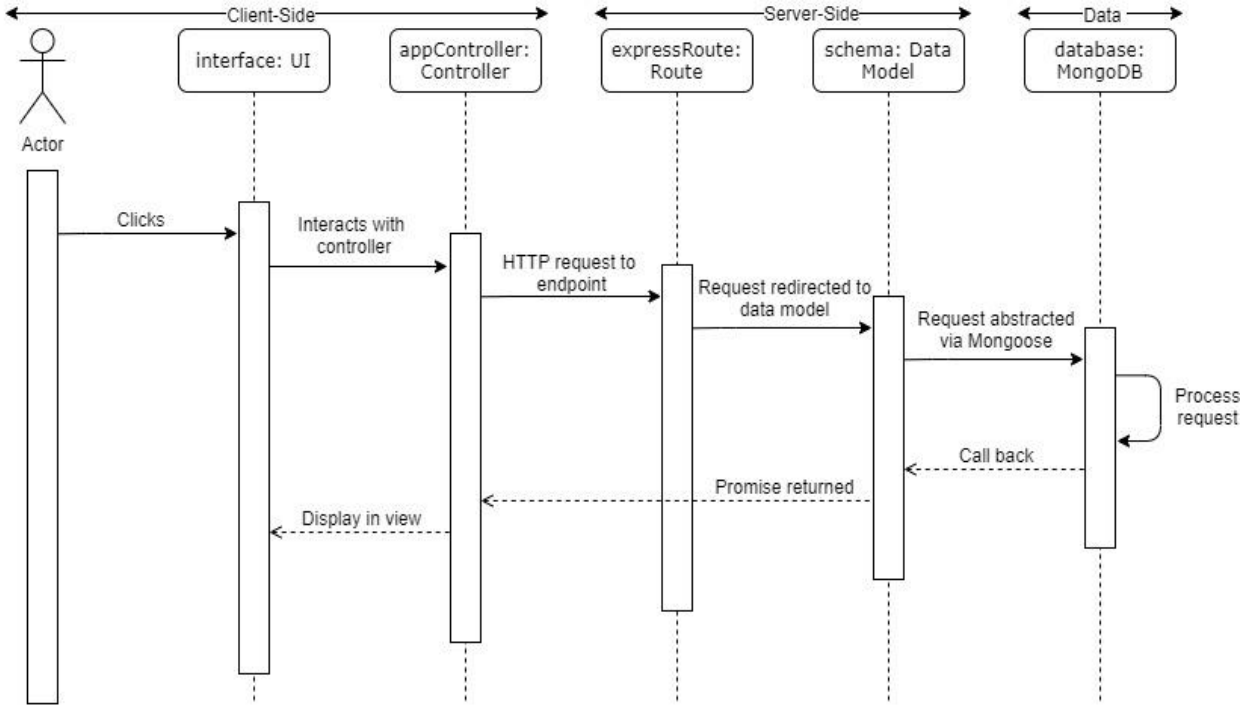


Figure 13. Sequence diagram for project.

15.16 User Interface Designs

15.17 TECHNOLOGY ANALYSIS

A full investigation was carried out at the start of the project to decide which technologies would be best suited for the development of this application.

15.17.1 Application

One of the aims of this project was for the application to be as compatible as possible, whilst still maintaining a responsive feel to the application. Due to the time constraints of the project, it was not feasible to develop a website, Android, and IOS application separately. Therefore, the only two options were to either develop a separate web application along with a cross-platform mobile application (using Microsoft Xamarin), or a progressive web application.

Separate Website and Mobile Application
Developing a separate website along with a cross-platform mobile application would require a huge amount of development time. In addition to this, using a platform such as Xamarin would be a completely new technology for the developer to learn which would take some time. Developing cross platform applications can also be difficult to prevent compatibility issues and ensure that the application works with the latest OS's. However, using native mobile applications instead of a web application does have some advantages. A native app would be much quicker to run on mobile devices as it would be optimised. Native application also has full access to the phone's hardware, which means that push notifications and other smart features can be implemented.

Progressive Web Application
Developing a progressive web application would take considerably less time and would mean that there would be more time to implement more features. Furthermore, the distribution of the application would be much easier, meaning that any user could access it via a browser. Progressive web applications have become much more advanced over the past few years and can even be installed on desktop, Android, and IOS via a web browser. This allows features such as, offline use and push notifications to be utilised.

15.17.2 Web Application

Angular
AngularJS uses TypeScript which, although is very similar to JavaScript, would be a new language for the developer to learn. AngularJS also has a steep learning curve, so time would have to be set aside for the developer to learn this. However, AngularJS does allow for two-way databinding, meaning that implementing the model-view-controller pattern would be easy.

ReactJS
ReactJS uses pure JavaScript which the developer is experienced in and would not need to learn. Although, it does use a component-based system which would be a new technology to learn but this does have the advantage of being about to re-use components throughout the application. ReactJS makes it very easy to create a progressive web application by using its 'create-react-app' functionality.

React Native
Reactive Native, much like ReactJS makes it very easy to build user interfaces that are both for mobile and desktop. It also allows for multiple platform compatibility to be implemented very easily. However, some expertise is needed it native mobile development to be able to use React Native to its full capability.

15.17.3 Middleware

NodeJS
NodeJS uses JavaScript which helps keep consistency throughout the project. It supports many packages, that are easy to install and use. Furthermore, NodeJS is very fast and can handle asynchronous requests whilst being easy to scale as the application increases in size.

.NET Core
.Net Core is very well documented and is supported by Microsoft. It is simple to use and lots of boilerplate code is written for you. However, it does have a steep learning curve and can have compatibility issues when trying to develop it not using a windows machine.

15.17.4 Database

When it came to the database, an investigation took place on whether it was best to use an SQL database or a NoSQL database.

SQL
An SQL database is well structured and defined using tables with keys and relationships. Complex queries can be created and used to manipulate data very easily. However, the entire schema should be defined at the start of the project and due to the relationships between tables, this can be difficult to modify later on.

NoSQL
A NoSQL database uses unstructured data and can be stored in JSON format. This makes it very easy to parse when the data is being pulled for the database. A NoSQL

database allow for easy scalability as they do not have to be pre-defined and schemas can be changed very easily without having any effect on the rest of the database.

15.17.5 Hosting

Heroku
Heroku makes deployment very easy and be integrated into a CI/CD pipeline without any issues. A microservice can be started within minutes to host a ReactJS and Node.js project. It is free to use as long as the server sleeps for at least 8 hours a day, as part of the 'hobby dyno'. However, Heroku does not provide much error logging or anyway to SSH into the server.

Google Firebase
Firebase has many services available; it offers static web hosting and database capabilities. It is also free to use for 'hobbyists' with certain limits on requests. However, after doing some research it does not look as easy to host a ReactJS and Node.js application compared to Heroku, which works out of the box.

Amazon Web Service (AWS)
AWS has many products available and a large amount of support online. However, the learning curve for AWS is steep and time would be needed to be spent to learn how it works. This would make it a bit of a challenge to Students are able to get a free $100 balance added to their account to use for hosting.

Microsoft Azure
Very similar to AWS, Azure has many products that can be used and also offers the same $100 credit for students. Despite this, it would only last a few months when hosting the cheapest server. Azure makes it very easy to setup servers, which are completely customisable in terms of hardware. However, like Azure there is a steep learning curve and it is mainly suitable for large businesses.

15.17.6 Links

Product	Link
Xamarin	https://dotnet.microsoft.com/apps/xamarin
Progressive Web Application	https://web.dev/progressive-web-apps/
AngularJS	https://angularjs.org/
Create-react-app	https://create-react-app.dev/
ReactJS	https://reactjs.org/
React Native	https://reactnative.dev/

Node.js	https://nodejs.org/en/
.NET Core	https://dotnet.microsoft.com/
MongoDB	https://www.mongodb.com/
Heroku	https://www.heroku.com/
Firebase	https://firebase.google.com/
AWS	https://aws.amazon.com/
Azure	https://azure.microsoft.com/en-gb/
Express	https://expressjs.com/
Cloudflare	https://www.cloudflare.com/

15.18 HOSTING

The web application is hosted on Heroku, running on their free dyno. This dyno allows for the application to be running for up to 16 hours a day; it has been set to sleep between 12am and 8am. The application is hosted on the following URL:

http://cloudpt.herokuapp.com/

Heroku allows for custom domains. The following domain was acquired and is serving the application:

https://cloudpt.me

The domain was purchased for free via Namecheap using GitHub's education package. Heroku does include an SSL package but they charge a monthly fee for it. Due to this, a Cloudflare DNS was set up to route all traffic via Cloudflare. This was set up because Cloudflare have SSL encryption setup free of charge, as well as the ability to view analytics of the website's requests and visitors. This can be seen in Figure 14.

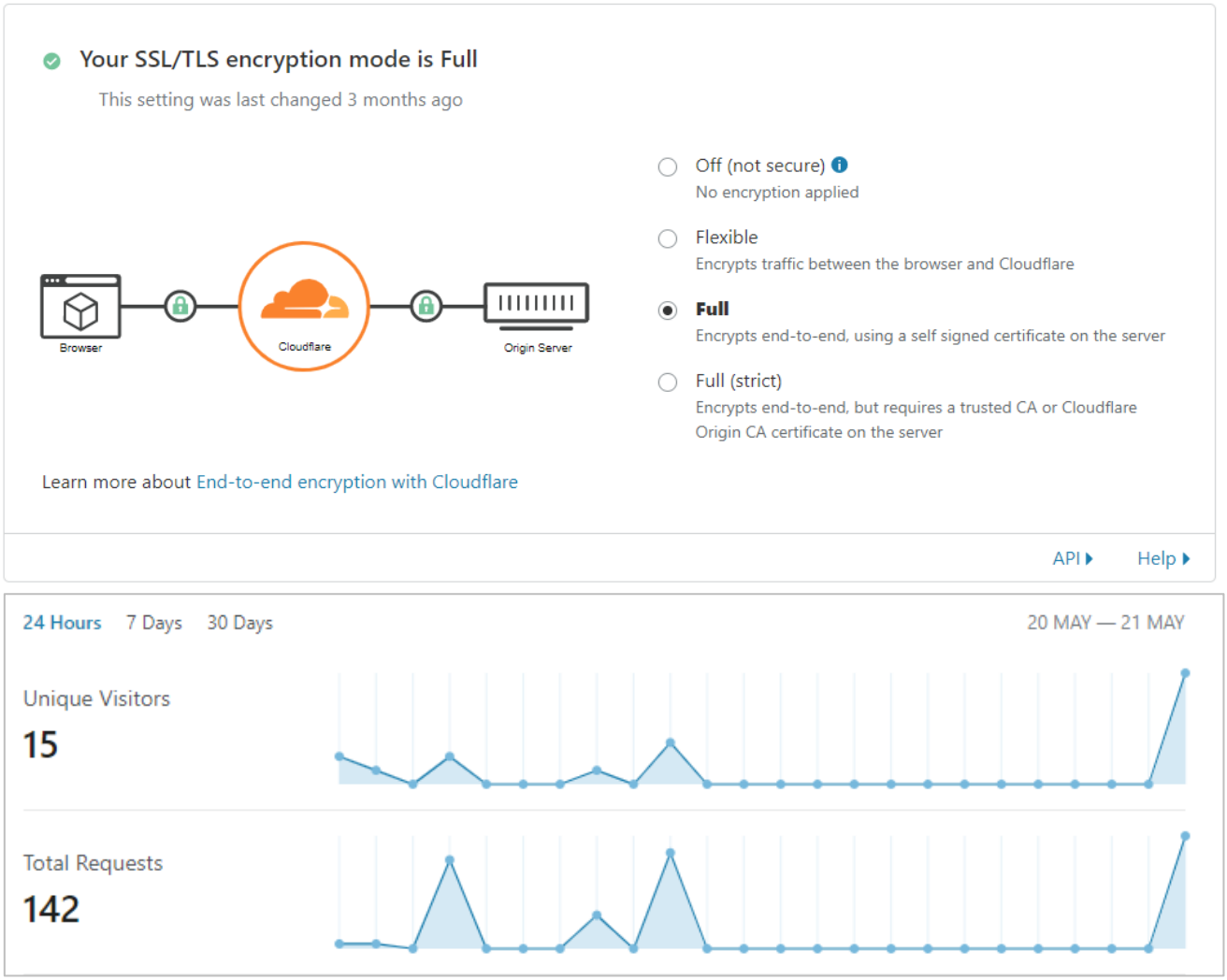


Figure 14. Cloudflare dashboard screenshots.

15.18.1 Links

Product	Link
Namecheap	https://www.namecheap.com/
Cloudflare	https://www.cloudflare.com/
GitHub Education	https://education.github.com/

15.19 API ROUTES

15.19.1 Authentication

Route	Method	Purpose
/api/login	POST	Attempts user login, redirects to another API route to be handled
/api/login/success	GET	Returns a true authentication if the login was successful
/api/login/failed	GET	Returns a false authentication if the login was not successful
/api/register	POST	Registers a new user
/api/auth	GET	Checks if a user is currently logged in
/api/user	GET	Returns current user
/api/user	POST	Returns user when provided with and email
/api/user/password	POST	Changes user password
/api/user/update	POST	Updates users' details (email, first name, last name)
/api/user/role	POST	Changes user role for PT to client or vice versa
/api/logout	GET	Logout user and destroy current session/cookie

15.19.2 Data

Route	Method	Purpose
/api/weight	GET	Gets all recorded weights for current user
/api/weight	POST	Saves a new weight for current user
/api/weight/:id	DELETE	Deletes a specific weight record
/api/height	GET	Gets users current height
/api/height	POST	Updates current user's height
/api/messages	GET	Gets all messages for the current user
/api/messages	POST	Stores new message that has been sent
/api/message/read	POST	Marks message as read
/api/workout/new	POST	Save a new workout that has been created
/api/workout	GET	Gets list of all workouts created by that user
/api/workout/:id	GET	Gets one specific workout
/api/workout/delete/:id	DELETE	Deletes a specific workout (marks it as not active)
/api/workout/update/:id	UPDATE	Updates an existing workout
/api/workout/assign	POST	Assigns a workout to a given user
/api/workout/assign	DELETE	Deletes an assignment of a workout of a given user
/api/workout/assigned	GET	Get all workouts assigned to current user
/api/workout/record	POST	Save a recorded workout
/api/workout/recorded	GET	Get all recorded workouts for current user
/api/user/relationship	GET	Get all active relationships for current user

/api/user/relationship	POST	Create a new relationship with current user and given user
/api/user/relationship	DELETE	Removes and active relationship between current user and given user

15.20 FUNCTIONALITY TESTING

Functionality testing of missing input data was not included in these tests as all required inputs used the 'required' HTML tag and forms are not able to be submitted without them. Tests have been broken into three sections:

- User Functionality Tests – Functionality that is used by all users
- Personal Trainer Functionality Tests – Functionality that is used by personal trainers
- Client Functionality Tests – Functionality that is used by clients

15.20.1 User Functionality Tests

Description	Steps	Input Data	Expected Result	Actual Result
Register attempt – Successful	1. Navigate to the register page 2. Input first name 3. Input last name 4. Input email 5. Input password 6. Click 'Register'	First Name: Dan Last Name: Thick Email: test@email.com Password: PasswordTest1234	Registration is successful and the user is re-directed to the login page.	As expected
Register attempt – Email already in use	1. Navigate to the register page 2. Input first name 3. Input last name 4. Input email 5. Input password 6. Click 'Register'	First Name: Dan Last Name: Thick Email: dan.thick@hotmail.co.uk Password: PasswordTest1234	Error message is displayed saying that the email is already in use.	As expected
Login attempt - Successful	1. Enter email address 2. Enter password 3. Click login button	Email: dan.thick@hotmail.co.uk Password: Password123	User will be logged in and home page is displayed.	As expected
Login attempt – Incorrect email	1. Enter incorrect email 2. Enter password 3. Click login button	Email: dan.thick@hotmail.co.uk Password: Password123	Message is displayed 'Incorrect email or password'.	As expected

Login attempt – Incorrect password	1. Enter email 2. Enter incorrect password 3. Click login button	Email: dan.thick@hotmail.co.uk Password: Password123	Message is displayed 'Incorrect email or password'.	As expected
Logout	1. Navigate to account page 2. Click 'Logout'	N/A	User is logged out and re-directed to application home page.	As expected
Change password – Successful	1. Navigate to account page 2. Click 'Change Password' 3. Enter current password 4. Enter new password 5. Enter new password confirmation 6. Click 'Change Password'	Current Password: Password1234 New Password: newPassword1234 Confirm New Password: newPassword1234	Message is displayed and password is changed successfully.	As expected
Change password – Current password incorrect	1. Navigate to account page 2. Click 'Change Password' 3. Enter incorrect current password 4. Enter new password 5. Enter new password confirmation 6. Click 'Change Password'	Current Password: wrongPassword New Password: newPassword1234 Confirm New Password: newPassword1234	Error message is displayed, and password is not changed.	As expected
Change password – New passwords do not match	1. Navigate to account page 2. Click 'Change Password' 3. Enter current password	Current Password: Password1234 New Password: newPassword1234 Confirm New Password: wrongPassword1234	Error message is displayed, and password is not changed.	As expected

	4. Enter new password 5. Enter new password confirmation that does not match 6. Click 'Change Password'			
Update details – Successful name change	1. Navigate to account page 2. Click 'Update Account Details' 3. Input changed name 4. Click 'Update Details'	First Name: Daniel Last Name: Thick	Success message is displayed, and users name has been changed.	As expected
Update details – Successful email change	1. Navigate to account page 2. Click 'Update Account Details' 3. Input changed email 4. Click 'Update Details'	Email: Dan.thick@gmail.com	Success message is displayed, and the users email address has been changed. User is then logged out. To then log in with new email address.	As expected
Update details – Successful name and email change	1. Navigate to account page 2. Click 'Update Account Details' 3. Input changed email and name 4. Click 'Update Details'	First Name: Daniel Last Name: Thick Email: Dan.thick@gmail.com	Success message is displayed, and the users email address has been changed. User is then logged out. To then log in with new email address.	As expected
Update details – Using an email that is already in use	1. Navigate to account page 2. Click 'Update Account Details' 3. Input changed email	Email: test@test.com	Error messages is displayed because email is already in use. No changes are made.	As expected

	4. Click ‘Update Details’			
Change user role	1. Navigate to account page 2. Click ‘Change User Role’ 3. Click ‘Change to Client/Personal Trainer’	N/A	Success message is show user role has been changed. User it logged out for changes to take effect.	As expected
Start new chat – Successful	1. Navigate to messages page 2. Click ‘Start New Chat’ 3. Enter email address 4. Click ‘Start Chat’	Email: john@smith.com	New chat window opens for the user that was inputted.	As expected
Start new chat – User does not exist	1. Navigate to messages page 2. Click ‘Start New Chat’ 3. Enter email address 4. Click ‘Start Chat’	Email: john@smith27.com	Error message appears because user does not exist.	As expected
Sending a message	1. Navigate to messages page 2. Either start new chat or open existing chat 3. Input message 4. Click ‘Send’	Message: Hello, how are you today?	Message is sent to the other user and message appears in chat window.	As expected

15.20.2 Personal Trainer User Functionality Tests

Description	Steps	Input Data	Expected Result	Actual Result
Adding a new client – Successful	1. Navigate to the home page 2. Click ‘Add Client’ 3. Input clients email address 4. Click ‘Add Client’	Email: john@smith.com	List is refreshed to show new client that has been added successfully.	As expected

Adding a new client – Email is not in use	1. Navigate to the home page 2. Click ‘Add Client’ 3. Input clients email address 4. Click ‘Add Client’	Email: john@smith27.com	Error message is displayed saying no user has that email.	As expected
Adding a new client – User is already a client of yours	1. Navigate to the home page 2. Click ‘Add Client’ 3. Input clients email address 4. Click ‘Add Client’	Email: john@smith.com	Error message is displayed saying that user is already one of your clients.	As expected
Adding a new client – User already has a personal trainer assigned to them	1. Navigate to the home page 2. Click ‘Add Client’ 3. Input clients email address 4. Click ‘Add Client’	Email: john@smith.com	Error message is displayed saying that the user already has a personal trainer assigned to them.	As expected
View client schedule	1. Navigate to the home page 2. Click ‘View Schedule’ for one of your clients	N/A	Clients schedule is displayed to the user.	As expected
Remove workout from client’s schedule	1. Navigate to the home page 2. Click ‘View Schedule’ for one of your clients 3. Click ‘Remove’ on the workout to be removed	N/A	Clients schedule list is refreshed and the selected workout is removed.	As expected
View client’s details	1. Navigate to the home page 2. Click ‘View Details’ for one of the clients	N/A	Clients details are displayed along with options for that client.	As expected

Remove user as client	1. Navigate to the home page 2. Click ‘View Details’ for one of the clients 3. Click ‘Remove Client’ 4. Click ‘OK’ on alert message box	N/A	Alert message box appears to confirm the user wants to remove the client	As expected
View client’s workout history	1. Navigate to the home page 2. Click ‘View Details’ for one of the clients 3. Click ‘View Workout History’	N/A	List of all the client’s workouts are displayed.	As expected
View specific workout	1. Navigate to the home page 2. Click ‘View Details’ for one of the clients 3. Click ‘View Workout History’ 4. Click ‘View Workout’ for any specific workout	N/A	Workout will be displayed, showing which exercises were completed or missed and any notes that were left.	As expected
Add exercise to new workout	1. Navigate to the workout page 2. Click ‘Create New Workout’ 3. Click ‘Add exercise’ 4. Choose exercise type 5. Choose a body part 6. Choose an exercise 7. Input weight 8. Input sets 9. Input repetitions	Exercise Type: Dumbbell Body Part: Chest Exercise: Incline Bench Press Weight: 20KG Sets: 5 Reps: 5	Exercise will appear in a list on the workout page.	As expected

	10. Click ‘Add’			
Delete exercise from workout	1. Click ‘Delete’ on a specific exercise within a workout	N/A	Exercise will be removed from the list.	As expected
Save new workout	1. Add exercise(s) to a new workout 2. Input workout name 3. Click ‘Save Workout’	Workout Name: Chest Workout	User will be redirect back to the main workout page and the new workout will appear in the list of workouts.	As expected
Edit workout	1. Navigate to the workout page 2. Click ‘Edit’ on a specific workout 3. Make necessary changes 4. Click ‘Save Workout’	N/A	User will be redirect to the main workout page and the workout will have been updated and saved.	As expected
Delete workout	1. Navigate to the workout page 2. Click ‘Delete’ on a specific workout	N/A	Workout list will refresh and workout will be removed.	As expected
Assign workout to client	1. Navigate to the workout page 2. Click ‘Assign’ 3. Choose a day of the week 4. Choose which clients to assign the workout to	Day of the week: Monday Client: John Smith	Message will show saying that workout has been assigned.	As expected

15.20.3 Client User Functionality Tests

Description	Steps	Input Data	Expected Result	Actual Result
Update user height	1. Navigate to the account page 2. Click ‘Update weight and height’	Height: 185cm	Page is refreshed and shows the updated height of the current user.	As expected

	3. Click ‘Update Height’ 4. Use slider to select current height 5. Click ‘Update Height’			
Update user weight	1. Navigate to the account page 2. Click ‘Update weight and height’ 3. Click ‘Update Weight’ 4. Use slider to select current weight 5. Click ‘Update Weight’	Weight: 80kg	Page is refreshed and shows the updated weight of the current user. As well as the graph updating to show the new weight.	As expected
Delete user weight	1. Navigate to the account page 2. Click ‘Update weight and height’ 3. Click the delete icon on a specific weight in list	N/A	Weight is removed. List and graph refresh to remove weight.	As expected
View workout history list	1. Navigate to the workout page 2. Click ‘Workout History’	N/A	List of all previously recorded workouts for the current user will be shown.	As expected
View specific workout	1. Navigate to the workout page 2. Click ‘Workout History’ 3. Click ‘View Workout’ on a specific workout	N/A	Workout will be displayed, showing which exercises were completed or missed and any notes that were left.	As expected
Start workout	1. Navigate to the workout page 2. Click ‘Start Workout’ in a specific workout	N/A	Workout is displayed with ability to mark exercises as	As expected

			completed or missed.	
Record workout	1. Navigate to the workout page 2. Click 'Start Workout' 3. Mark all exercises as completed or missed 4. Input optional note 5. Click 'Finish Workout'	Notes: Workout went well today!	Message is displayed saying that workout was completed. Workout is saved and can now be viewed in workout history.	As expected

15.21 USABILITY TESTING

Usability testing was conducted towards the end of the project to get an understanding as to how users would interact with the application. It also allowed users to give feedback and help make any adjustments that would improve the application.

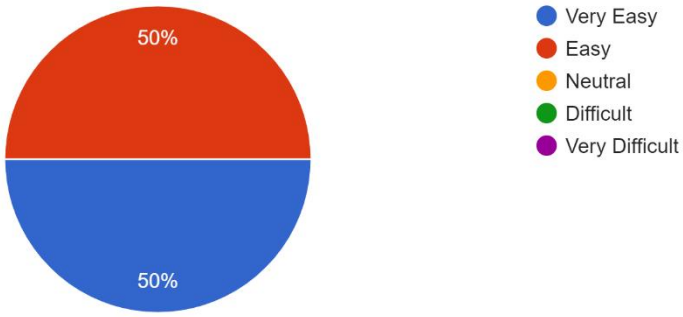
15.21.1 Tasks

Task	**Input**
Register yourself as new user and log in to the application.	Name: *Your name* Email: Your email Are you a PT? *Yes* Password: *Any password*
Start a new chat with a user and send a message.	User to message: dan.thick@hotmail.co.uk Message: Hello, how are you?
Add a new client.	Email: john@smith.com
Create a new workout with just one exercise.	Workout Name: Leg Day Exercise: Barbel Squat 50kg 5 sets of 5 repetitions
Assign the workout that was just created to John Smith on a Tuesday.	N/A

15.21.2 Results

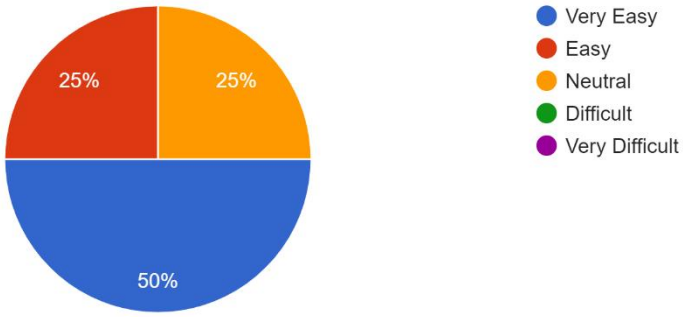
How easy was it to register a new account and login?

4 responses



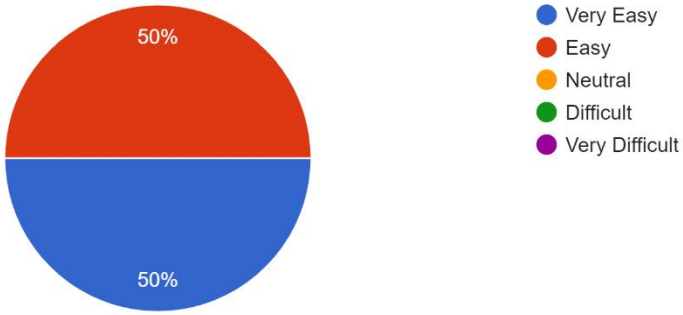
How easy was it to start a new chat and send a message?

4 responses



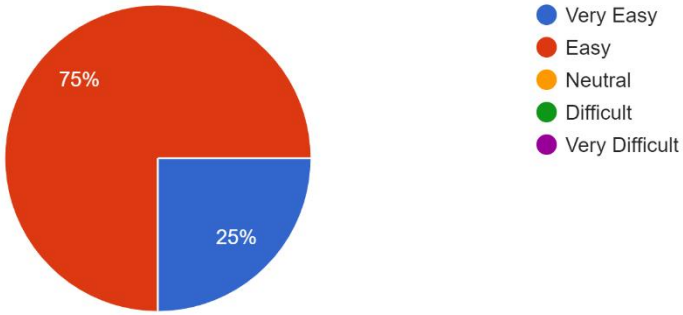
How easy was it to add a new client?

4 responses



How easy was it to create a new workout?

4 responses



How easy was it to assign a workout to a client?

4 responses

Overall, the results of the usability testing were very positive, and all participants found all the tasks easy. Participants were also able to add a comment about what they thought would improve the functionality or design. One of the comments that was suggests was to make information on screen clearer and could be broken into sections. This was improved by implemented a ‘card’ based frontend for lists and other pages with lots of information on screen at once. This can be seen in figure 15.

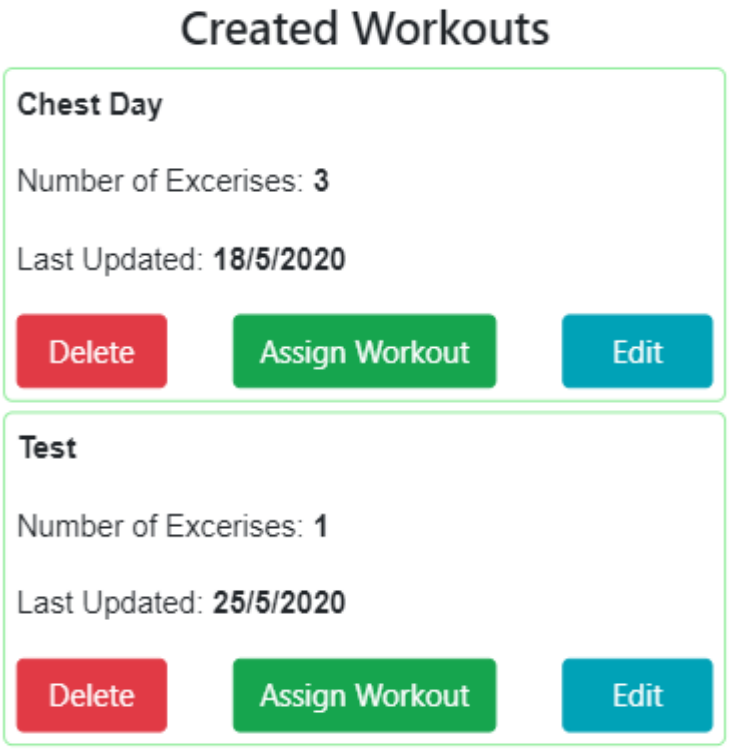


Figure 15. Screenshot from CloudPT.

15.22 Unit Tests

Two libraries were used to complete testing on the middleware of the application, they were Mocha and Chai. Mocha provides the structure of the tests and allows the tests to have a ‘describe’ and ‘it’ part. This helps when tests are printed to the console as it easily shows which tests and passed or failed. Chai is being used to send requests to the middleware and check the responses. These two libraries working together can be seen in figure 16.

```
describe("Register User", () => {
    it("should register a new user", (done) => {
        agent
            .post("/api/register")
            .type('form')
            .send(user)
            .end((err, res) => {
                res.should.have.status(200);
                done();
            });
    });
});
```

Figure 16. Example unit test for middleware.

To test ReactJS components, the libraries Jest, and Enzyme are needed. Jest provides the ‘describe’ and ‘it’ structure to the tests, whilst Enzyme allows for a shallow copy of components to be made for them to be tested.

```
describe('Testing account component', () => {
    it('renders without crashing', () => {
        shallow(<Account />);
    });
});
```

Figure 17. Example unit test for frontend.

During the build process of the application, both sets of tests for the frontend and middleware are ran. The CI/CD pipeline will only push the build to production if all the tests are passed. Figure 18 shows the terminal output of the tests. Due to time constraints of the project there was only a limited number of unit tests implemented. In the future my unit tests should be implemented to have a much higher percent of code coverage.

```
  Testing all account components render without crashing
    √ testing messages component (41ms)
    √ testing account component (1ms)
    √ testing edit user details component (2ms)
    √ testing weight and height component (3ms)
    √ testing change password component (1ms)
    √ testing changing user role component (1ms)
    √ testing login component (3ms)
    √ testing register component (1ms)
  Testing all personal trainer components render without crashing
    √ testing homepage component (3ms)
    √ testing workout page component (4ms)
  Testing all client components render without crashing
    √ testing workout history component (2ms)
    √ testing homepage component (2ms)
    √ testing workout component (2ms)

Test Suites: 1 passed, 1 total
Tests:       15 passed, 15 total
```

Figure 18. Example terminal output for unit tests.