# CA472 Final Documentation

## Executive Summary

Club-Kit is a sports club admin application. Our goal is take a significant workload off of the hardworking volunteers in sports clubs everywhere. We do this through implementing online features such as player registration, renting pitches, sending out training rosters and a club shop.

Local clubs rely on volunteers to give up their free time to assist in keeping a club running smoothly on a daily basis. Volunteers jobs can vary from managing teams to completing administrative work and everything in between. Club-Kit uses advanced online systems to help automate daily tasks and in-turn, dramatically reduce the workload of each volunteer.

The player registrations feature removes the need for clubs to host a registration night at the beginning of the year, as most clubs still do and will help improve and speed up the process of registrations and avoid potential errors especially handling cash payments. The system will also keep track of who is registered correctly and who is fully payed up.

Renting out facilities and pitches can be one of a clubs primary sources of income. When facilities are not in use by teams within the club, the club has the option to rent the facilities to third parties online through a secure booking system. While most clubs already rent their facilities, it is commonly completed over the phone and by a volunteer constantly taking phone calls and noting who has booked and when. Our online system allows for users to look up available slots within the club, book and pay online.

The club shop feature allows clubs to advertise and sell club jerseys, shorts, hats and other types of club gear online. Users can purchase online from the club shop, pay online and collect their purchase from the club. Club admins will be responsible for populating club shop by including photos, titles, categories, sizes, prices, number of units and descriptions of the goods.

Allowing for clubs to easily schedule and show training rosters online is another feature of Club-Kit. This features allows players and parents quickly find training rosters on their clubs page.

A problem with sports clubs is that volunteers become tired of the constant work needed to keep a club running. This leads to people walking away and leaving the club short of volunteers. Our aim is to make running a club as easy as possible for everyone. Each feature is available as a standalone solution or as a full package which allows for Club-Kit to tailor for each individual club’s needs.

# CONOR WARD

Student

## PROFILE

I am an Enterprise Computing Student at Dublin City University. I am working towards the end of my degree and hope to secure a job in some form of software engineering.

I am currently working the application Club-Kit for my final year project.

## CONTACT

**PHONE:**

0860403000

**WEBSITE:**

Club-kit.ie

**EMAIL:**

Conor.ward27@mail.dcu.ie

## HOBBIES

Football

Gym

Software Developing

## EDUCATION

**Dublin City University**

September 2014 – May 2019
Enterprise Computing Student

**Ratoath College**

September 2009 – June 2014
Secondary School student receiving 400 points in the leaving cert

## WORK EXPERIENCE

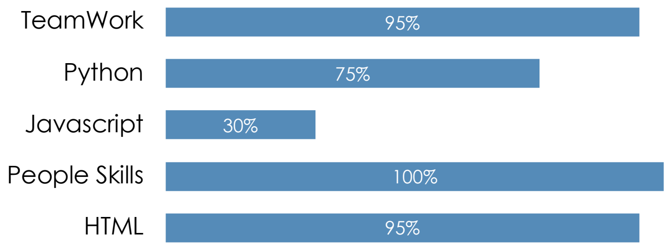
**Department of Agriculture, Intern Web developer**

January 2018–September 2018
Internship as part of college course. Worked on department websites including coford.ie, agriappeals.gov.ie and agriculture.gov.ie

**BoyleSports, Cashier - Part time**

September 2018–Present
Cashier Duties

## SKILLS



# GAVIN BOYLE

Student

## PROFILE

I am an Enterprise Computing Student at Dublin City University. I have just completed my final examinations awaiting results.

I am currently working part-time for UPS in a software development team whilst working on an application called Club-Kit as part of my final year project.

## CONTACT

**PHONE:**

0863098579

**WEBSITE:**

Club-kit.ie

**EMAIL:**

Gavin.Boyle8@mail.dcu.ie

## HOBBIES

Software Developing

Cooking

Football

Travel

Gym

## EDUCATION

**Dublin City University**

September 2014 – May 2019
Enterprise Computing Student

**Ratoath College**

September 2008 – June 2014
Secondary School Student

## WORK EXPERIENCE

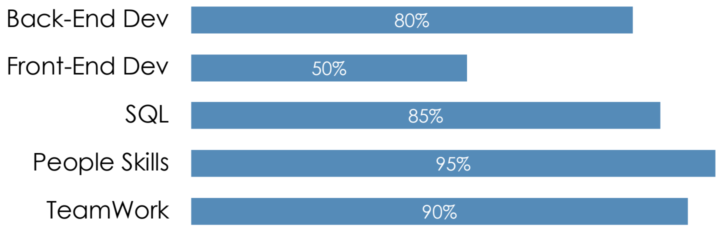
**Nightline Logistics - (Customer Services, Receptionist, Dispatch Team, Warehouse Work)**

July 2012 – March 2018
I've worked in a vast variety of roles within Nightline Logistics group since the age of 16.

**Nightline Logistics/UPS - (Data Analyst, IT Support, Automation)**

March 2018 – Present
Began working as part of a college internship, continued part-time since joining.

## SKILLS



## Organizational Structure

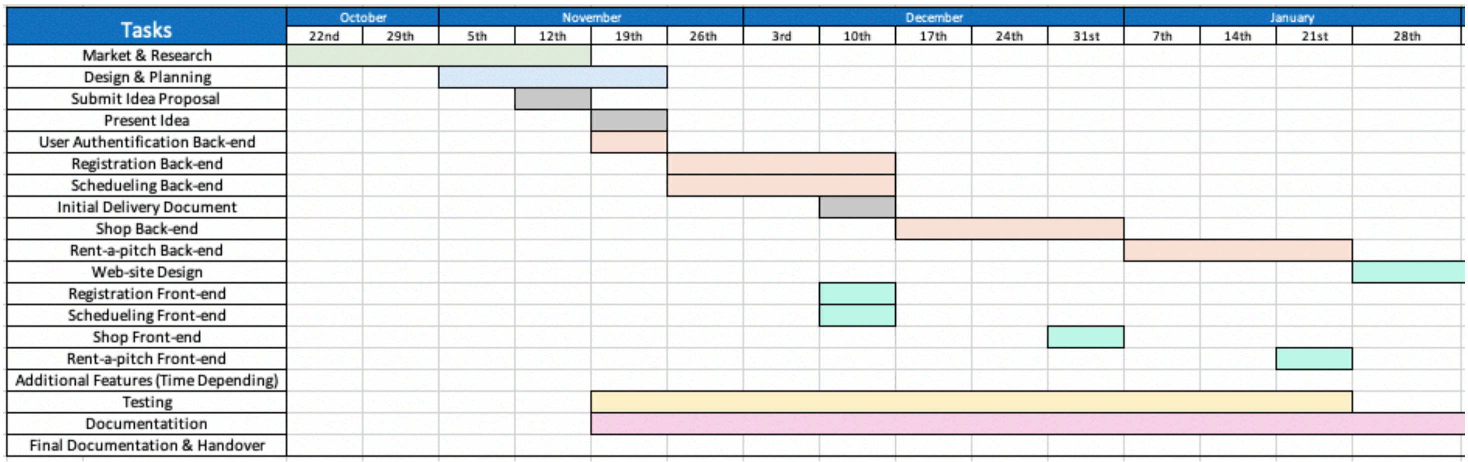
Gavin Boyle and Conor Ward – Co-developers

## Blogs

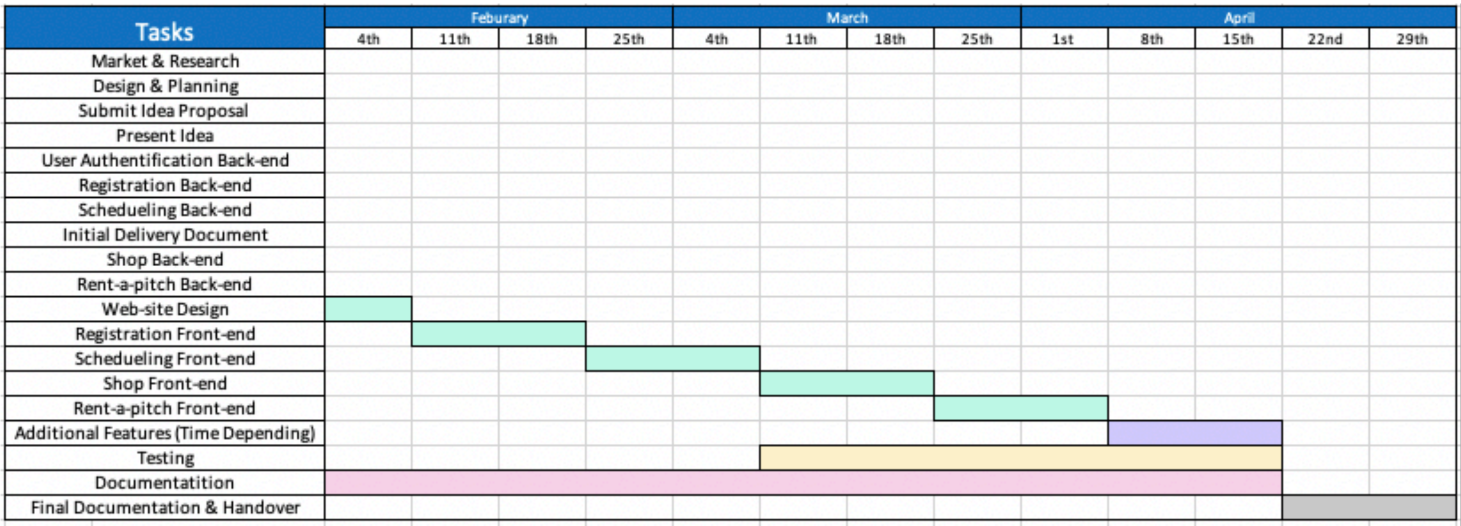
Gavin Boyle - https://blogs.computing.dcu.ie/wordpress/boyleg8/

Conor Ward - https://blogs.computing.dcu.ie/wordpress/wardc27/

## Time Description



| Tasks | October | | November | | | | December | | | | | January | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 22nd | 29th | 5th | 12th | 19th | 26th | 3rd | 10th | 17th | 24th | 31st | 7th | 14th | 21st | 28th |
| Market & Research | ■ | ■ | ■ | ■ | | | | | | | | | | | |
| Design & Planning | | | ■ | ■ | ■ | | | | | | | | | | |
| Submit Idea Proposal | | | | ■ | | | | | | | | | | | |
| Present Idea | | | | | ■ | | | | | | | | | | |
| User Authentification Back-end | | | | | ■ | | | | | | | | | | |
| Registration Back-end | | | | | | ■ | ■ | ■ | | | | | | | |
| Schedueling Back-end | | | | | | ■ | ■ | ■ | | | | | | | |
| Initial Delivery Document | | | | | | | | ■ | | | | | | | |
| Shop Back-end | | | | | | | | | ■ | ■ | ■ | | | | |
| Rent-a-pitch Back-end | | | | | | | | | | | | ■ | ■ | ■ | |
| Web-site Design | | | | | | | | | | | | | | | ■ |
| Registration Front-end | | | | | | | | ■ | | | | | | | |
| Schedueling Front-end | | | | | | | | ■ | | | | | | | |
| Shop Front-end | | | | | | | | | | | ■ | | | | |
| Rent-a-pitch Front-end | | | | | | | | | | | | | | ■ | |
| Additional Features (Time Depending) | | | | | | | | | | | | | | | |
| Testing | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | |
| Documentatition | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Final Documentation & Handover | | | | | | | | | | | | | | | |



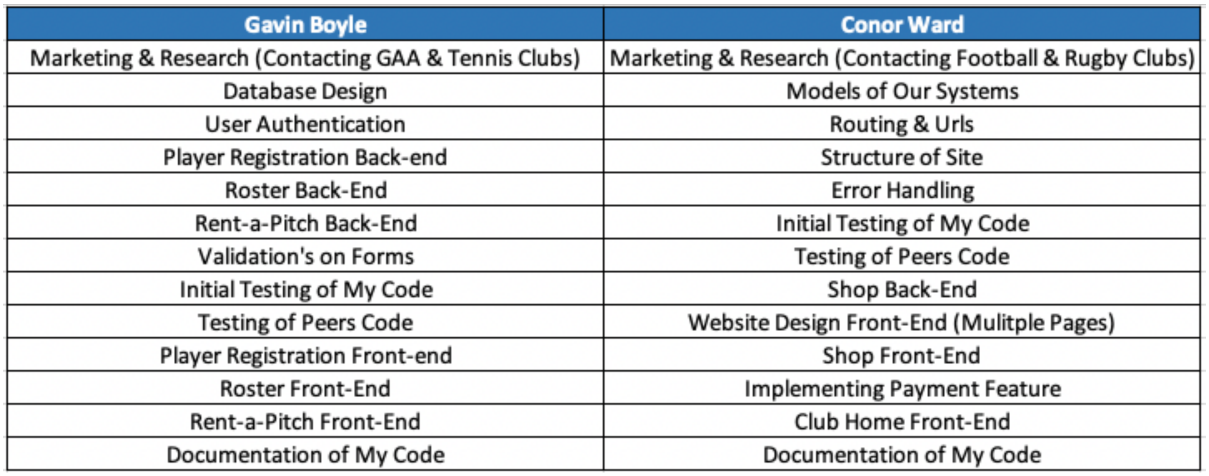
| Tasks | Feburary | | | | March | | | | April | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4th | 11th | 18th | 25th | 4th | 11th | 18th | 25th | 1st | 8th | 15th | 22nd | 29th |
| Market & Research | | | | | | | | | | | | | |
| Design & Planning | | | | | | | | | | | | | |
| Submit Idea Proposal | | | | | | | | | | | | | |
| Present Idea | | | | | | | | | | | | | |
| User Authentification Back-end | | | | | | | | | | | | | |
| Registration Back-end | | | | | | | | | | | | | |
| Schedueling Back-end | | | | | | | | | | | | | |
| Initial Delivery Document | | | | | | | | | | | | | |
| Shop Back-end | | | | | | | | | | | | | |
| Rent-a-pitch Back-end | | | | | | | | | | | | | |
| Web-site Design | ■ | | | | | | | | | | | | |
| Registration Front-end | | ■ | ■ | | | | | | | | | | |
| Schedueling Front-end | | | | ■ | ■ | | | | | | | | |
| Shop Front-end | | | | | | ■ | ■ | | | | | | |
| Rent-a-pitch Front-end | | | | | | | | ■ | ■ | | | | |
| Additional Features (Time Depending) | | | | | | | | | | ■ | ■ | | |
| Testing | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | | |
| Documentatition | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | |
| Final Documentation & Handover | | | | | | | | | | | | ■ | ■ |

Our Gantt chart shows that we did the web site design over a couple of weeks. This ensured that the site was made to a high standard and would display correctly for users on all types of devices. The various front ends were also completed over a few weeks to ensure that the user interface of each of the features was appealing, efficient and easy to use for our users.

We planned to complete the development of the application towards the start of April, this turned out to not be possible as we had to balance other aspects of college as well as this project. Testing of the front-end commenced as we started to get most of the design completed.

This gave us plenty of time to ensure everything was done correctly and up to a high standard. Testing of the application and documentation for the application was done continuously over the timeline of the project. The final documentation and handover of the application will be over the weeks of May 13th and 20th unlike what we originally planned to be April 22nd and 29th.

The workload was distributed between the two team members, Gavin Boyle and Conor Ward. Tasks were completed separately but in most cases we helped each other through difficult tasks. The workload is distributed as follows:



| Gavin Boyle | Conor Ward |
|---|---|
| Marketing & Research (Contacting GAA & Tennis Clubs) | Marketing & Research (Contacting Football & Rugby Clubs) |
| Database Design | Models of Our Systems |
| User Authentication | Routing & Urls |
| Player Registration Back-end | Structure of Site |
| Roster Back-End | Error Handling |
| Rent-a-Pitch Back-End | Initial Testing of My Code |
| Validation's on Forms | Testing of Peers Code |
| Initial Testing of My Code | Shop Back-End |
| Testing of Peers Code | Website Design Front-End (Mulitple Pages) |
| Player Registration Front-end | Shop Front-End |
| Roster Front-End | Implementing Payment Feature |
| Rent-a-Pitch Front-End | Club Home Front-End |
| Documentation of My Code | Documentation of My Code |

**Business Case**

**Detailed description**

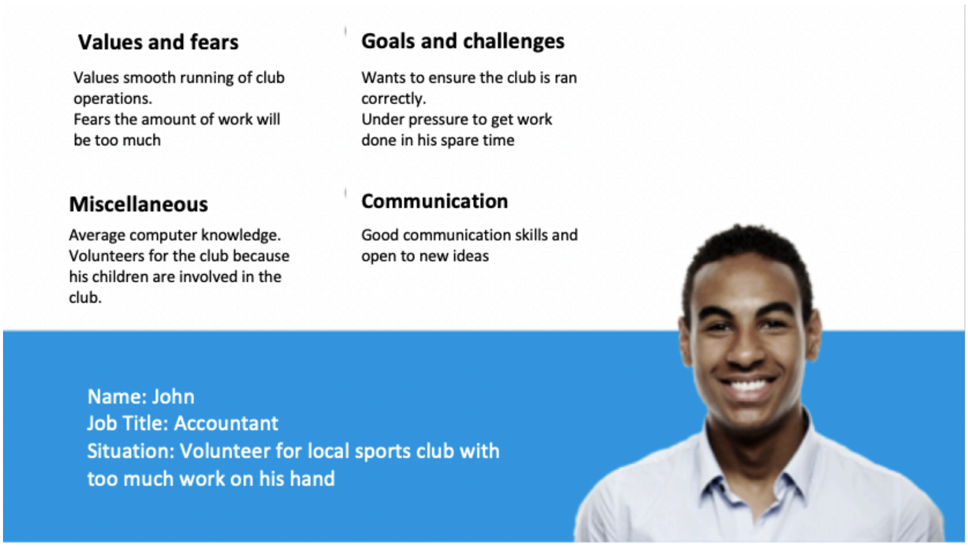
Club-Kit is an online system where sports clubs can complete day to day administrative tasks online to ensure the smooth running of business operations and club operations within the club. Clubs pay a yearly fee for the service in which they choose which solutions they would like for their club. The system allows clubs to move towards a modern approach to club administration and allows for club volunteers and admins to reduce their workload in their club, allowing for a better work-life balance while volunteering at their club.

**Value analysis**

The users of the application will be primarily club admins who use the application to complete clubs administrative tasks. Other users include players, parents and managers who will be able to view club information and use features such as rent a pitch, training rosters and the club shop.

The application helps club admins and sports clubs by allowing tasks to be easily completed online and reducing the workload of unpaid volunteers. The application solves the problem of volunteers being overwhelmed by the amount of work involved in running a club removing the risk of volunteers walking away from their duties.

### User Personae



**Values and fears**

Values smooth running of club operations.
Fears the amount of work will be too much

**Goals and challenges**

Wants to ensure the club is ran correctly.
Under pressure to get work done in his spare time

**Miscellaneous**

Average computer knowledge.
Volunteers for the club because his children are involved in the club.

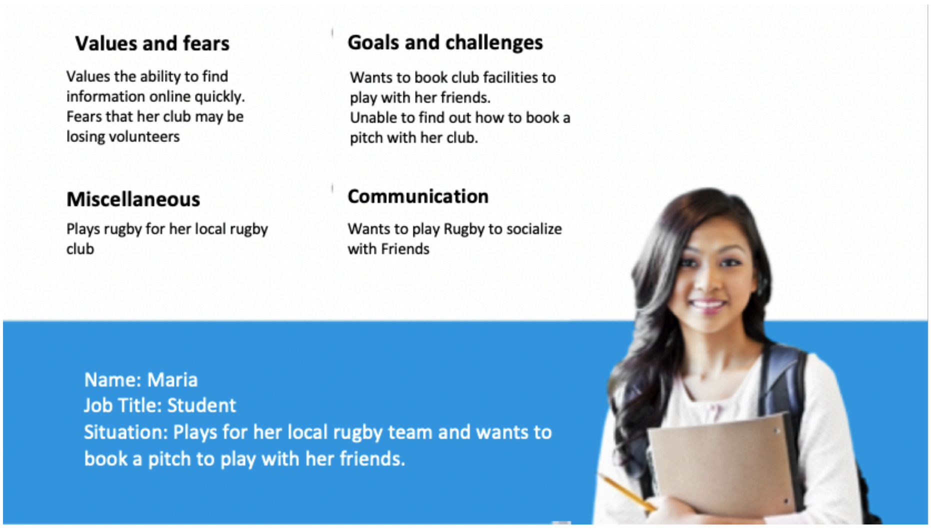
**Communication**

Good communication skills and open to new ideas

Name: John
Job Title: Accountant
Situation: Volunteer for local sports club with too much work on his hand

Above is an example persona of a person volunteering for a sports club. John is an accountant who volunteers for his local football club. His children are members of the club, so he wants to be involved and offer to help the club in his spare time.

John sometimes finds that there is too much work to be completed each week and is considering leaving the club. He feels that the way the club runs is not very efficient and is open to ideas on how to improve how the club is run. John is an example of our target customers. Our application could help with tasks volunteers like himself and other volunteers are having to complete each week.



**Values and fears**

Values the ability to find information online quickly.
Fears that her club may be losing volunteers

**Goals and challenges**

Wants to book club facilities to play with her friends.
Unable to find out how to book a pitch with her club.

**Miscellaneous**

Plays rugby for her local rugby club

**Communication**

Wants to play Rugby to socialize with Friends

Name: Maria
Job Title: Student
Situation: Plays for her local rugby team and wants to book a pitch to play with her friends.

Above is an example persona of the typical user of the application. Maria is a member of her local Rugby club and is looking to book a pitch for her friends to play on in their spare time. She is unable to find out how to do this as the club doesn't advertise pitch booking online or have any information about the club online. With our system the club will have the ability to rent their pitches online, handle payments for registered and un-registered members to easily find and book.

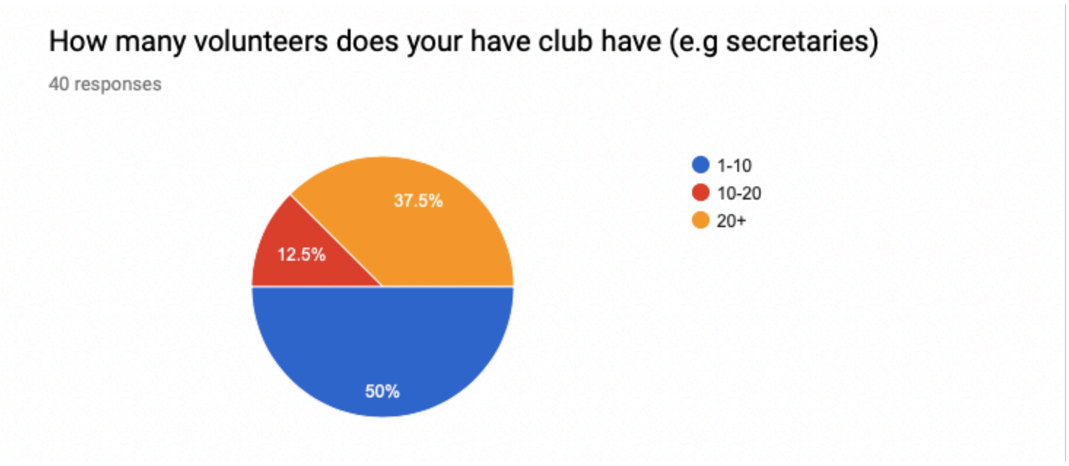
## Market Analysis

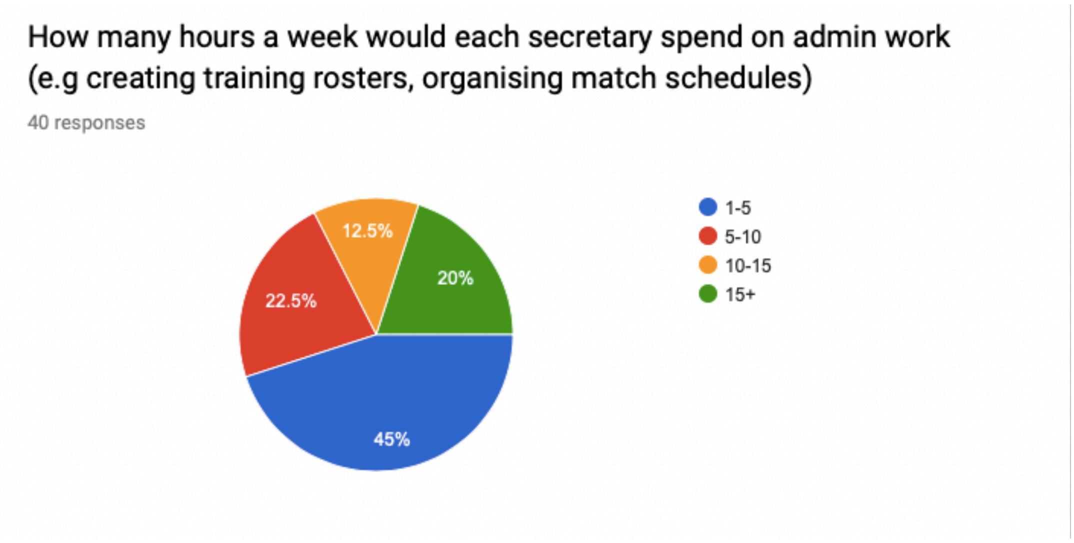
The market consists solely of sports clubs around the country and in the future, international clubs. Clubs of any sport will be able to utilize the application including soccer, GAA, rugby, tennis and various other sports. The clubs who will use this application will be looking to speed up and improve their administrative tasks which keep the club running.

Currently, most clubs continue to operate in a traditional manner, take for example member registration, some clubs host registration nights in order to collect membership payments, other methods include members paying managers who then have the responsibility to handling cash and safely passing it onto club admins for them to handle.

It's not just registrations that are still done in an old fashioned way. The ways in which clubs are booking pitches, creating and sending out training rosters and selling club items are all stuck in the past. This is seen from first-hand experience as well as market research we have completed.

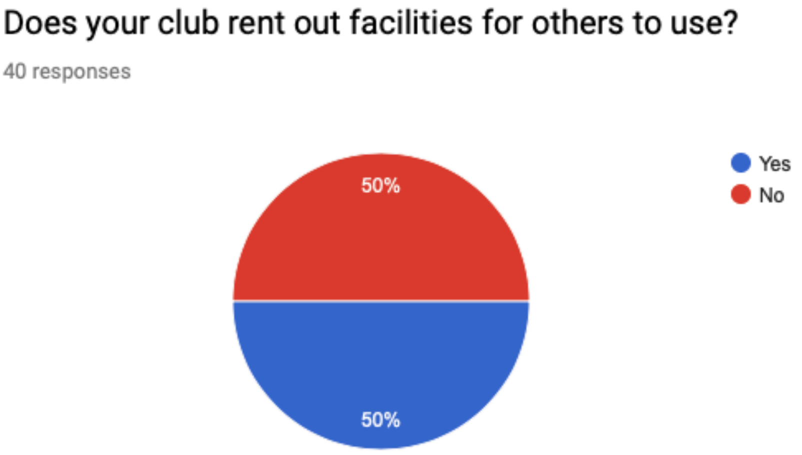
We conducted some market research in which we contacted a random selection of sports clubs, ranging from various sports and sizes. This survey consisted of questions about each club, how they currently operate and if they would be interested in using a system which they can operate club tasks online. We received a good response to the survey and it showed that the majority of clubs would be interested in our features.





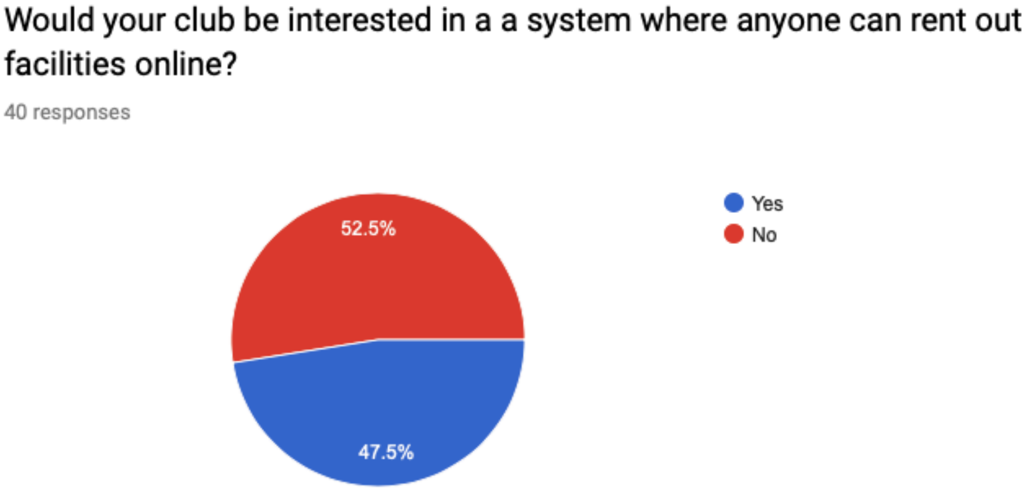
How many hours a week would each secretary spend on admin work (e.g creating training rosters, organising match schedules)

40 responses



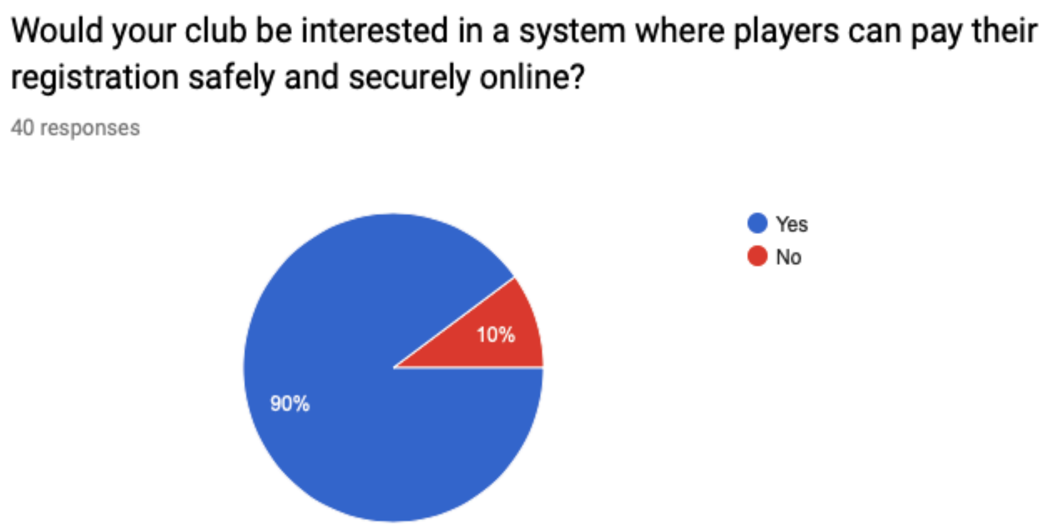
Does your club rent out facilities for others to use?

40 responses



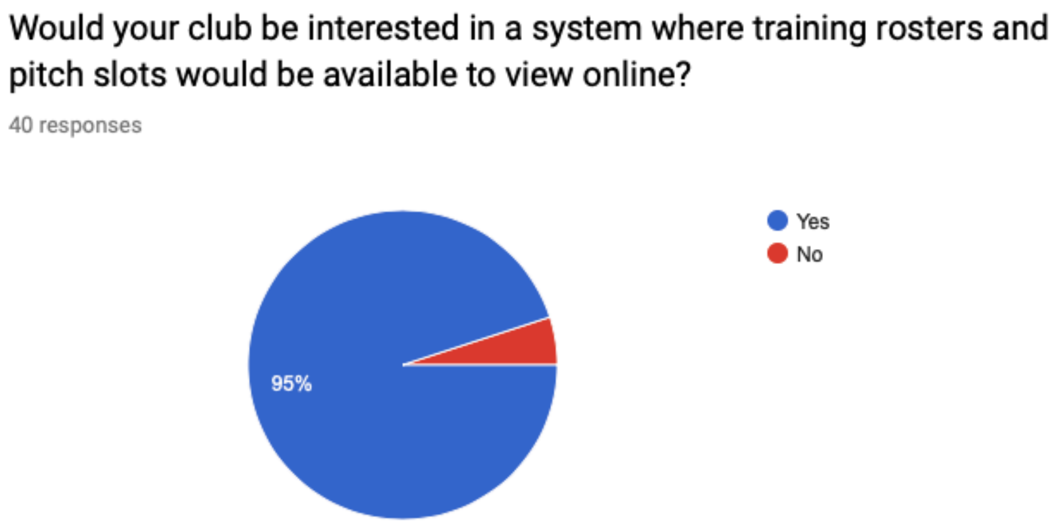
Would your club be interested in a a system where anyone can rent out facilities online?

40 responses



Would your club be interested in a system where players can pay their registration safely and securely online?

40 responses



Would your club be interested in a system where training rosters and pitch slots would be available to view online?

40 responses

These results show that clubs are interested in training rosters online and a safe online registration system but not as many clubs actually rent out their facilities. This made us rethink our business plan and we came to the solution that each individual club will be able to purchase packages with features on the system that they feel they need. This ensures clubs are only paying for what they need.

We also gained an understanding that each club has a different number of volunteers who all work various hours. The survey also includes responses for each club on how they think we could improve the system even further. One response was to include a feature for clubs to sell their club gear and items. This was something we had not thought of previously and we felt it was a feature that we had to include.

We gained an understanding that not all clubs carried out these processes in the same way, for example upon researching different types of registration forms used by individual clubs we came to realise that they collect all different data which can be challenging to cater for all these different types. For our prototype we just focused on the one type, with the hope to incorporate more in the future.

### Key resources

For this application to become commercially viable, it would require various resources such as capital, manpower, marketing and a suitable infrastructure to deploy it on. A prototype application has already been developed  and with such resources as more capital, the prototype can be improved and made ready for real-world, everyday use. With more people added to the team they will assist in the further research, development, marketing and more.

Marketing will be an important aspect of this applications commercial viability. Currently, there are already a few applications with similar features available online. The problem that these applications are having is that not many people have heard of them. That is why for the release of this application, we will have a strong marketing team to show users who and what we are so that we will gain a competitive advantage over rival applications.

### Costs

Certain costs we would incur would include finding a suitable infrastructure to deploy and host the application on. Cloud providers will charge for a set amount of traffic and data each month to use their service while some other cloud providers charge based on a pay as you use model, meaning we only pay for what our application uses each month. It could also be beneficial to look at having our own in-house infrastructure to host the application. This could be quite costly as it requires purchasing servers and other equipment.

### Revenue Streams

Our proposed revenue streams will be from each club paying a yearly fee to use our system. Upon sign-up, clubs will have the option to choose which features they would like to purchase. This ensures that clubs are only paying for what they feel is necessary for their club.

Clubs will be able to sign-up to the application free of charge. Without purchasing any packages, they will be able to create a club page where they can add information like club details, teams in the club and pitches associated to their club.

When clubs do sign-up for the free option, advertisements will be included on the right side of their homepage. These advertisements will be very minimal but allow for us as a business to generate funds even from clubs who do not pay for our application. We feel that once the club signs up to use the free features of the application they will be enticed to avail of the various other features we offer.

## Pricing Model

### Player registration feature

This part of the application allows clubs to invite members to pay their registrations fees and complete their registration forms online. This feature is available to clubs for €100 per year on a subscription basis. Once the club has purchased this feature it will be immediately available to them.

### Training rosters feature

This feature is allowing clubs to display their training rosters in an online platform that is easily accessible to members of the club. Clubs will be able to purchase this feature for €75 per year on a subscription basis.

### Rent-a-pitch feature

The rent-a-pitch feature allows clubs to rent their pitches to users online. Clubs can choose which pitches they would like to rent out and will be able to show which time slots are available. This feature is available for €100 per year on a subscription basis.

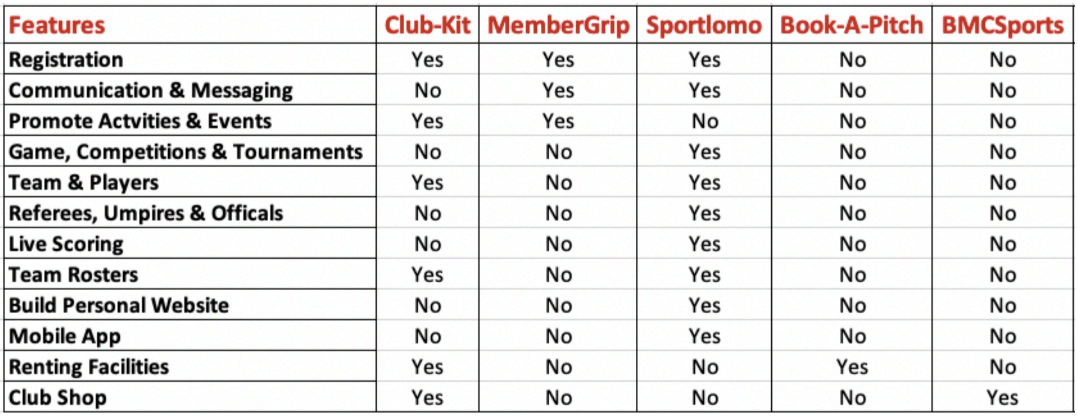
### Shop feature

Clubs will have the opportunity to sell club jerseys and gear online. They will be able to upload items in which users can pay for online. Users will then be able to collect the item at the club’s grounds. This feature is available for €150 per year on a subscription basis.

We offer a discounted price of €400 per year when a club purchases all of these features in the one package. Clubs will be billed once yearly for the full amount of which packages they have chosen. Clubs will be able to upgrade their choices at any time and will pay the yearly subscription rate at the time of upgrade. Clubs also have the ability to cancel or downgrade their package, this will result in the automatic subscription being cancelled and the packages will still be in use up until the time of expiry from when the packages were initially purchased.

### Competitor Profiles

There are a few competitors already in the market in terms of an online club application. These include SportLomo, Book-a-Pitch and BMCSports. These competitors all have their own unique way of implementing features similar to ours. The difference between Club-Kit and these various competitors, is that our application is an all-in-one solution. The differences between these key competitors can be seen in the table below.



| Features | Club-Kit | MemberGrip | Sportlomo | Book-A-Pitch | BMCSports |
|---|---|---|---|---|---|
| Registration | Yes | Yes | Yes | No | No |
| Communication & Messaging | No | Yes | Yes | No | No |
| Promote Actvities & Events | Yes | Yes | No | No | No |
| Game, Competitions & Tournaments | No | No | Yes | No | No |
| Team & Players | Yes | No | Yes | No | No |
| Referees, Umpires & Officals | No | No | Yes | No | No |
| Live Scoring | No | No | Yes | No | No |
| Team Rosters | Yes | No | Yes | No | No |
| Build Personal Website | No | No | Yes | No | No |
| Mobile App | No | No | Yes | No | No |
| Renting Facilities | Yes | No | No | Yes | No |
| Club Shop | Yes | No | No | No | Yes |

### Sportlomo

Sportlomo is arguably our biggest rival as they offer the most features in their application. They include features such as registrations, messaging, referee support and live scoring. The problem with Sportlomo is that they only offer these features under a single price plan. This means that for those smaller clubs that won't utilize all of these features, it's a waste of their money. While at Club-Kit, we offer clubs the chance to choose which features they want to pay for.

### Book-a-pitch

Book-a-pitch is an application which essentially only has one feature, booking club pitches. They allow for clubs to sign-up and advertise their pitches for rent on the book-a-pitch website. This feature is available at a cost of €250 per annum. That is only €150 cheaper than all of our features combined.

### BMCSports

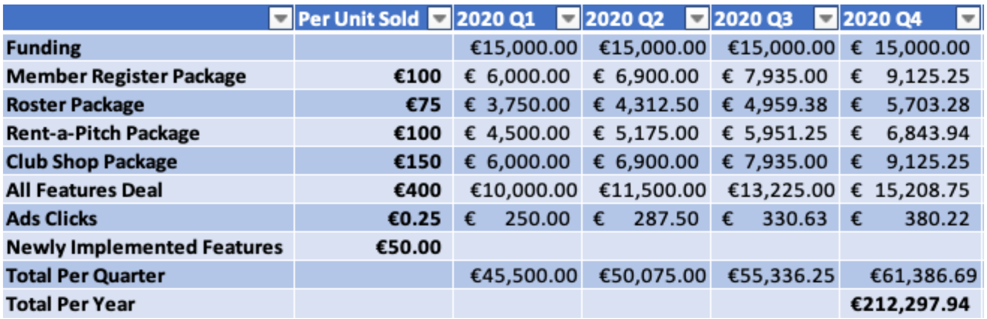
BMCSports offers sports clubs the opportunity to sell their club gear online. Clubs request a quote for BMCSports to sell and deliver their items. Club-Kit offers this feature along with many others for an up-front subscription fee.

Our competitive advantage is that we offer all of these features as standalone packages or as an all in one solution. This allows for clubs to choose which features they want to pay for and will encourage clubs to switch from one of our competitors to our solution.

# 3 Year Financial Projection

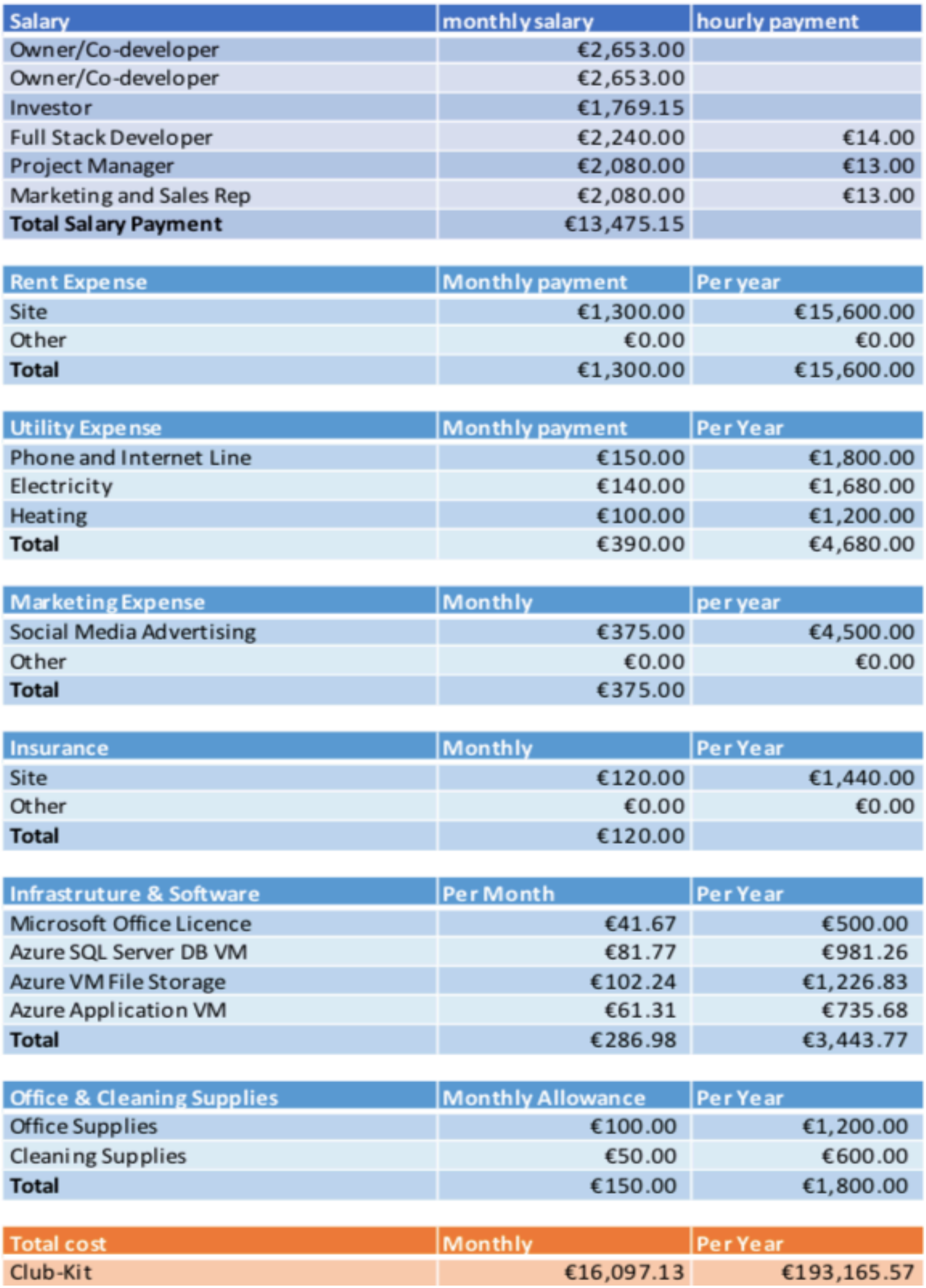
## Year 1

### Income



| | Per Unit Sold | 2020 Q1 | 2020 Q2 | 2020 Q3 | 2020 Q4 |
|---|---|---|---|---|---|
| **Funding** | | €15,000.00 | €15,000.00 | €15,000.00 | € 15,000.00 |
| **Member Register Package** | **€100** | € 6,000.00 | € 6,900.00 | € 7,935.00 | € 9,125.25 |
| **Roster Package** | **€75** | € 3,750.00 | € 4,312.50 | € 4,959.38 | € 5,703.28 |
| **Rent-a-Pitch Package** | **€100** | € 4,500.00 | € 5,175.00 | € 5,951.25 | € 6,843.94 |
| **Club Shop Package** | **€150** | € 6,000.00 | € 6,900.00 | € 7,935.00 | € 9,125.25 |
| **All Features Deal** | **€400** | €10,000.00 | €11,500.00 | €13,225.00 | € 15,208.75 |
| **Ads Clicks** | **€0.25** | € 250.00 | € 287.50 | € 330.63 | € 380.22 |
| **Newly Implemented Features** | **€50.00** | | | | |
| **Total Per Quarter** | | €45,500.00 | €50,075.00 | €55,336.25 | €61,386.69 |
| **Total Per Year** | | | | | **€212,297.94** |

### Expenses



| Salary | monthly salary | hourly payment |
|---|---|---|
| Owner/Co-developer | €2,653.00 | |
| Owner/Co-developer | €2,653.00 | |
| Investor | €1,769.15 | |
| Full Stack Developer | €2,240.00 | €14.00 |
| Project Manager | €2,080.00 | €13.00 |
| Marketing and Sales Rep | €2,080.00 | €13.00 |
| **Total Salary Payment** | €13,475.15 | |

| Rent Expense | Monthly payment | Per year |
|---|---|---|
| Site | €1,300.00 | €15,600.00 |
| Other | €0.00 | €0.00 |
| **Total** | €1,300.00 | €15,600.00 |

| Utility Expense | Monthly payment | Per Year |
|---|---|---|
| Phone and Internet Line | €150.00 | €1,800.00 |
| Electricity | €140.00 | €1,680.00 |
| Heating | €100.00 | €1,200.00 |
| **Total** | €390.00 | €4,680.00 |

| Marketing Expense | Monthly | per year |
|---|---|---|
| Social Media Advertising | €375.00 | €4,500.00 |
| Other | €0.00 | €0.00 |
| **Total** | €375.00 | |

| Insurance | Monthly | Per Year |
|---|---|---|
| Site | €120.00 | €1,440.00 |
| Other | €0.00 | €0.00 |
| **Total** | €120.00 | |

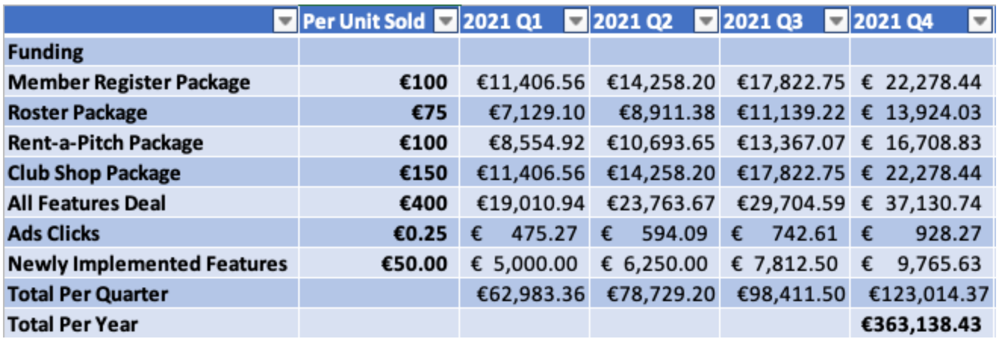
| Infrastruture & Software | Per Month | Per Year |
|---|---|---|
| Microsoft Office Licence | €41.67 | €500.00 |
| Azure SQL Server DB VM | €81.77 | €981.26 |
| Azure VM File Storage | €102.24 | €1,226.83 |
| Azure Application VM | €61.31 | €735.68 |
| **Total** | €286.98 | €3,443.77 |

| Office & Cleaning Supplies | Monthly Allowance | Per Year |
|---|---|---|
| Office Supplies | €100.00 | €1,200.00 |
| Cleaning Supplies | €50.00 | €600.00 |
| **Total** | €150.00 | €1,800.00 |

| Total cost | Monthly | Per Year |
|---|---|---|
| Club-Kit | €16,097.13 | €193,165.57 |

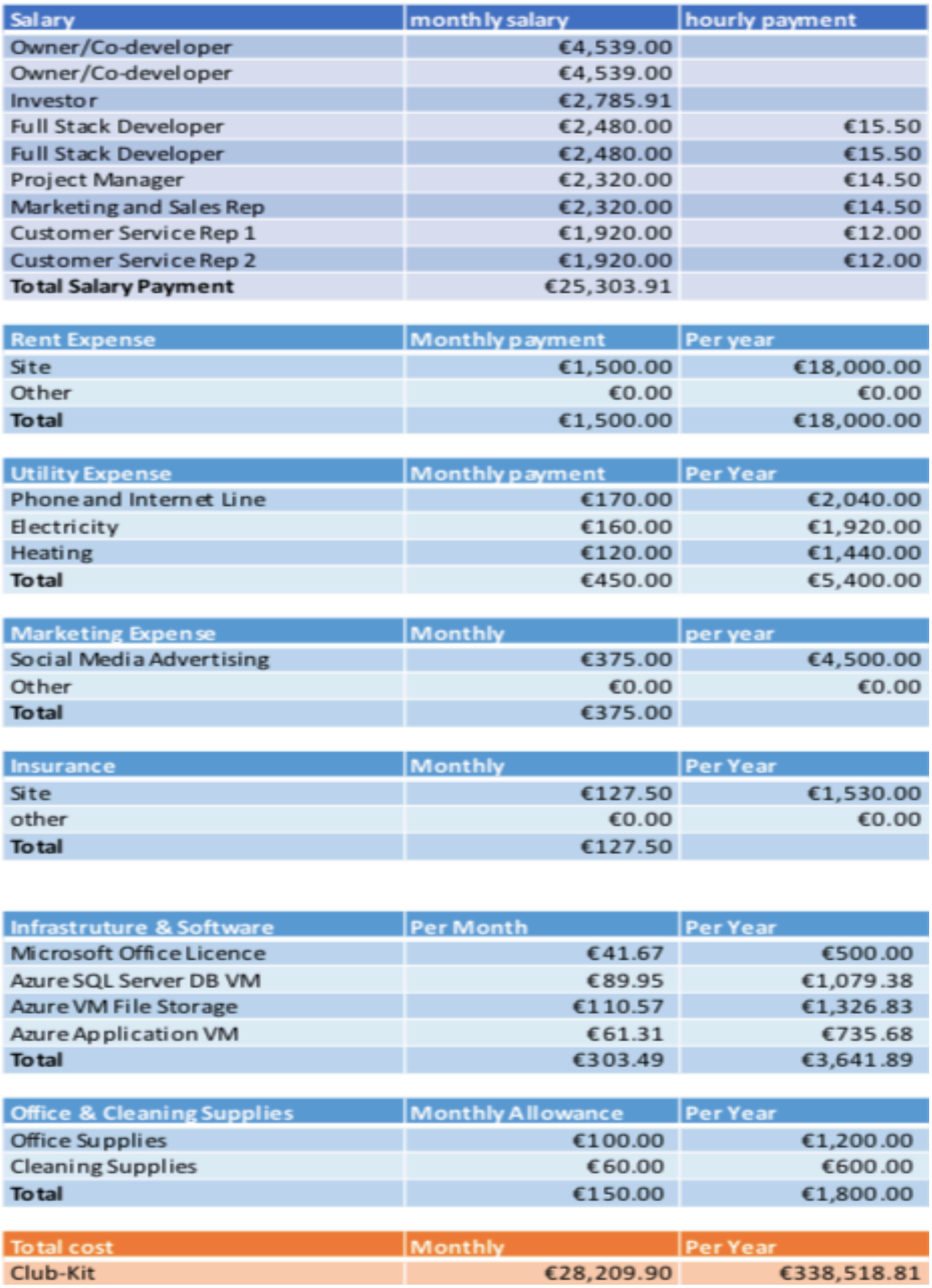
## Year 2

### Income



| | Per Unit Sold | 2021 Q1 | 2021 Q2 | 2021 Q3 | 2021 Q4 |
|---|---|---|---|---|---|
| Funding | | | | | |
| Member Register Package | €100 | €11,406.56 | €14,258.20 | €17,822.75 | € 22,278.44 |
| Roster Package | €75 | €7,129.10 | €8,911.38 | €11,139.22 | € 13,924.03 |
| Rent-a-Pitch Package | €100 | €8,554.92 | €10,693.65 | €13,367.07 | € 16,708.83 |
| Club Shop Package | €150 | €11,406.56 | €14,258.20 | €17,822.75 | € 22,278.44 |
| All Features Deal | €400 | €19,010.94 | €23,763.67 | €29,704.59 | € 37,130.74 |
| Ads Clicks | €0.25 | € 475.27 | € 594.09 | € 742.61 | € 928.27 |
| Newly Implemented Features | €50.00 | € 5,000.00 | € 6,250.00 | € 7,812.50 | € 9,765.63 |
| Total Per Quarter | | €62,983.36 | €78,729.20 | €98,411.50 | €123,014.37 |
| Total Per Year | | | | | €363,138.43 |

### Expenses



| Salary | monthly salary | hourly payment |
|---|---|---|
| Owner/Co-developer | €4,539.00 | |
| Owner/Co-developer | €4,539.00 | |
| Investor | €2,785.91 | |
| Full Stack Developer | €2,480.00 | €15.50 |
| Full Stack Developer | €2,480.00 | €15.50 |
| Project Manager | €2,320.00 | €14.50 |
| Marketing and Sales Rep | €2,320.00 | €14.50 |
| Customer Service Rep 1 | €1,920.00 | €12.00 |
| Customer Service Rep 2 | €1,920.00 | €12.00 |
| Total Salary Payment | €25,303.91 | |

| Rent Expense | Monthly payment | Per year |
|---|---|---|
| Site | €1,500.00 | €18,000.00 |
| Other | €0.00 | €0.00 |
| Total | €1,500.00 | €18,000.00 |

| Utility Expense | Monthly payment | Per Year |
|---|---|---|
| Phone and Internet Line | €170.00 | €2,040.00 |
| Electricity | €160.00 | €1,920.00 |
| Heating | €120.00 | €1,440.00 |
| Total | €450.00 | €5,400.00 |

| Marketing Expense | Monthly | per year |
|---|---|---|
| Social Media Advertising | €375.00 | €4,500.00 |
| Other | €0.00 | €0.00 |
| Total | €375.00 | |

| Insurance | Monthly | Per Year |
|---|---|---|
| Site | €127.50 | €1,530.00 |
| other | €0.00 | €0.00 |
| Total | €127.50 | |

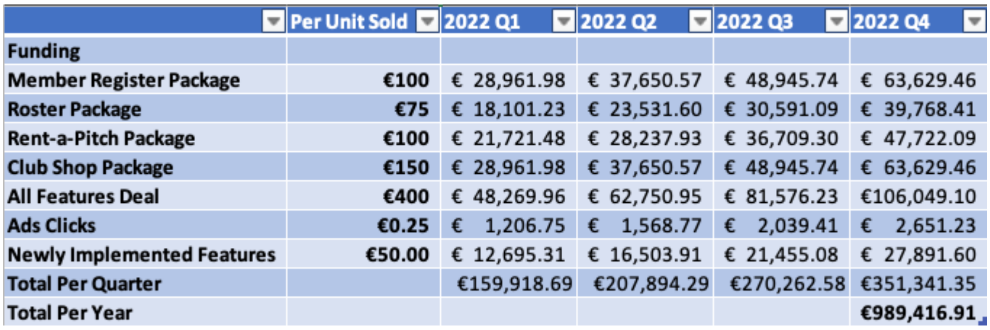
| Infrastruture & Software | Per Month | Per Year |
|---|---|---|
| Microsoft Office Licence | €41.67 | €500.00 |
| Azure SQL Server DB VM | €89.95 | €1,079.38 |
| Azure VM File Storage | €110.57 | €1,326.83 |
| Azure Application VM | €61.31 | €735.68 |
| Total | €303.49 | €3,641.89 |

| Office & Cleaning Supplies | Monthly Allowance | Per Year |
|---|---|---|
| Office Supplies | €100.00 | €1,200.00 |
| Cleaning Supplies | €60.00 | €600.00 |
| Total | €150.00 | €1,800.00 |

| Total cost | Monthly | Per Year |
|---|---|---|
| Club-Kit | €28,209.90 | €338,518.81 |

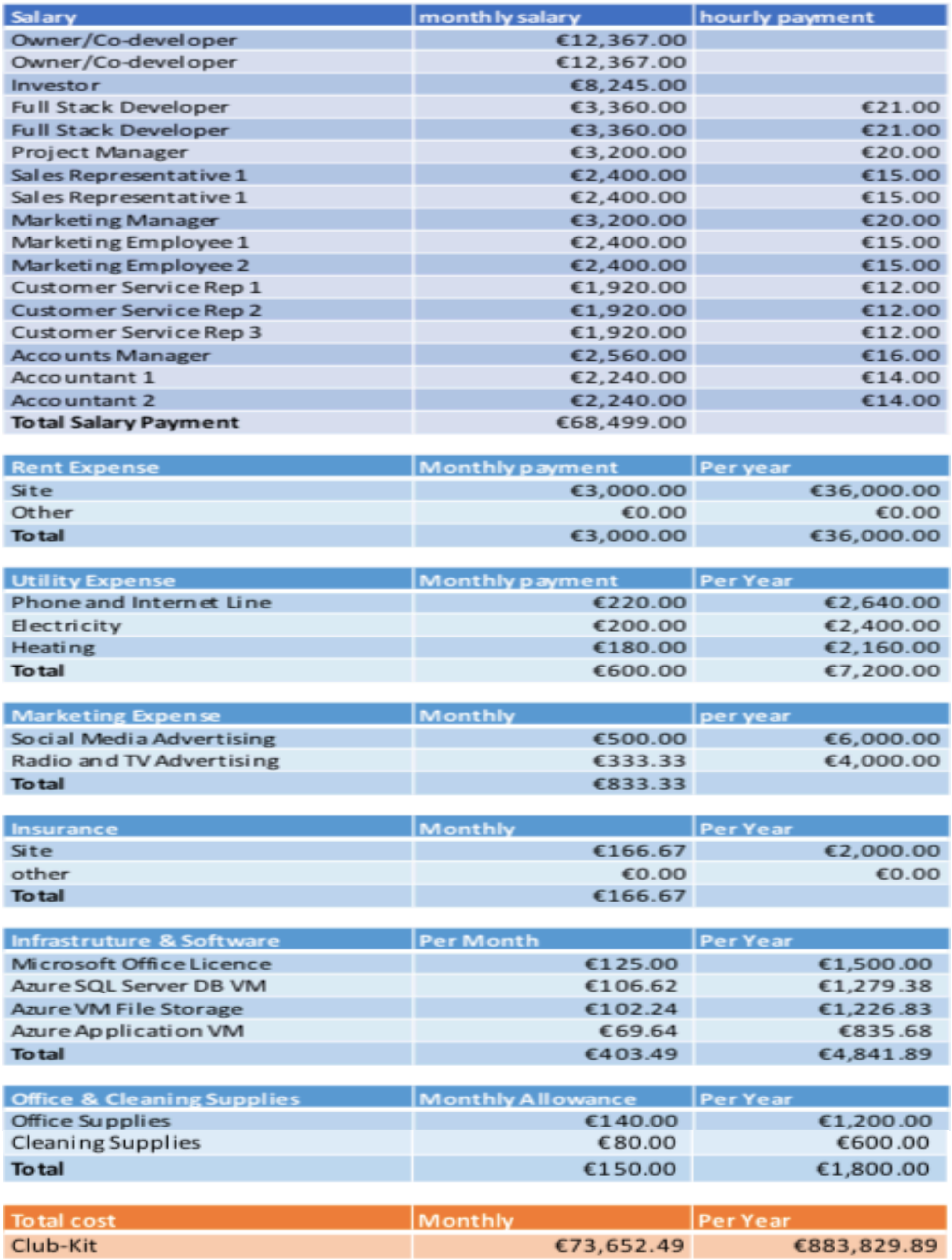
## Year 3

## Income



| | Per Unit Sold | 2022 Q1 | 2022 Q2 | 2022 Q3 | 2022 Q4 |
|---|---|---|---|---|---|
| Funding | | | | | |
| Member Register Package | €100 | € 28,961.98 | € 37,650.57 | € 48,945.74 | € 63,629.46 |
| Roster Package | €75 | € 18,101.23 | € 23,531.60 | € 30,591.09 | € 39,768.41 |
| Rent-a-Pitch Package | €100 | € 21,721.48 | € 28,237.93 | € 36,709.30 | € 47,722.09 |
| Club Shop Package | €150 | € 28,961.98 | € 37,650.57 | € 48,945.74 | € 63,629.46 |
| All Features Deal | €400 | € 48,269.96 | € 62,750.95 | € 81,576.23 | €106,049.10 |
| Ads Clicks | €0.25 | € 1,206.75 | € 1,568.77 | € 2,039.41 | € 2,651.23 |
| Newly Implemented Features | €50.00 | € 12,695.31 | € 16,503.91 | € 21,455.08 | € 27,891.60 |
| Total Per Quarter | | €159,918.69 | €207,894.29 | €270,262.58 | €351,341.35 |
| Total Per Year | | | | | €989,416.91 |

## Expenses



| Salary | monthly salary | hourly payment |
|---|---|---|
| Owner/Co-developer | €12,367.00 | |
| Owner/Co-developer | €12,367.00 | |
| Investor | €8,245.00 | |
| Full Stack Developer | €3,360.00 | €21.00 |
| Full Stack Developer | €3,360.00 | €21.00 |
| Project Manager | €3,200.00 | €20.00 |
| Sales Representative 1 | €2,400.00 | €15.00 |
| Sales Representative 1 | €2,400.00 | €15.00 |
| Marketing Manager | €3,200.00 | €20.00 |
| Marketing Employee 1 | €2,400.00 | €15.00 |
| Marketing Employee 2 | €2,400.00 | €15.00 |
| Customer Service Rep 1 | €1,920.00 | €12.00 |
| Customer Service Rep 2 | €1,920.00 | €12.00 |
| Customer Service Rep 3 | €1,920.00 | €12.00 |
| Accounts Manager | €2,560.00 | €16.00 |
| Accountant 1 | €2,240.00 | €14.00 |
| Accountant 2 | €2,240.00 | €14.00 |
| Total Salary Payment | €68,499.00 | |

| Rent Expense | Monthly payment | Per year |
|---|---|---|
| Site | €3,000.00 | €36,000.00 |
| Other | €0.00 | €0.00 |
| Total | €3,000.00 | €36,000.00 |

| Utility Expense | Monthly payment | Per Year |
|---|---|---|
| Phone and Internet Line | €220.00 | €2,640.00 |
| Electricity | €200.00 | €2,400.00 |
| Heating | €180.00 | €2,160.00 |
| Total | €600.00 | €7,200.00 |

| Marketing Expense | Monthly | per year |
|---|---|---|
| Social Media Advertising | €500.00 | €6,000.00 |
| Radio and TV Advertising | €333.33 | €4,000.00 |
| Total | €833.33 | |

| Insurance | Monthly | Per Year |
|---|---|---|
| Site | €166.67 | €2,000.00 |
| other | €0.00 | €0.00 |
| Total | €166.67 | |

| Infrastruture & Software | Per Month | Per Year |
|---|---|---|
| Microsoft Office Licence | €125.00 | €1,500.00 |
| Azure SQL Server DB VM | €106.62 | €1,279.38 |
| Azure VM File Storage | €102.24 | €1,226.83 |
| Azure Application VM | €69.64 | €835.68 |
| Total | €403.49 | €4,841.89 |

| Office & Cleaning Supplies | Monthly Allowance | Per Year |
|---|---|---|
| Office Supplies | €140.00 | €1,200.00 |
| Cleaning Supplies | €80.00 | €600.00 |
| Total | €150.00 | €1,800.00 |

| Total cost | Monthly | Per Year |
|---|---|---|
| Club-Kit | €73,652.49 | €883,829.89 |

## Overview of 3 Year Financial Projection

In the first quarter of 2020 we hope to start with approximately 60 customers using our registration package, 50 customers using our roster package, 45 customers using our Rent-a-Pitch package, 40 customers using our shop package and 25 customers using our all in one solution. We have also integrated advertisements into pages of clubs who are suing the free version of the application. Its estimated that we will receive approximately €0.25 per click through rate. It is estimated that growth and popularity of the application will grow by about 15% each quarter in 2020. We hope to receive €60,000 in funding from an investor for 10% of the business' shares. We estimate our total income to come to €212,297.94 and our expenses including salaries, rent, paying for an infrastructure to host our application and various other costs amounts to €193,165.57. This brings the company to a total profit of €19,132.37 in our first year while paying back investment funding from an investor.

In year 2 we hope to continue this steady growth of the business by implementing new features into the application. We estimate an income growth of approximately 25% each quarter in 2021. It will be possible to hire more staff to keep the business growing, move to a slightly better location and upgrade all of our cloud infrastructure. We estimate to pull in about €363,138.43 in income in year 2 with expenses coming to a total of €338,518.81. This brings the company to a total profit of €24,619.62 in our second year.

Year 3 is when we plan to begin moving our business international. There is only a finite amount of clubs in Ireland who we give our business to and going international opens us up to a lot more potential customers. We estimate that our growth will continue by approximately 30%. This is possible because we have invested more into our marketing while we are expanding. More employees have been hired to ensure smooth running of everyday business operations we will have moved to an even bigger office to facilitate everyone. We estimate an income of €989,416.91 and expenses of €883,829.89. This brings the company to a total profit of €105,587.02 in our third year.

The two owners and co-developers, Conor Ward and Gavin Boyles salaries are estimated on 15% of the companies income. The investor give the company €60,000 over the first 4 quarters and in return will receive 10% of the company's shares.

# Table of Content

# 1. Introduction

## 1.1 Overview

The platform being designed is best illustrated as an online sports club management system which both admins and members can avail of. The system allows sports clubs to sign up for free and manage a club home page to share information and posts, handle team, pitch and membership information. Additional paid services include member registration, scheduling/rostering, rental of pitches and a fully customizable shop to handle in club purchases.

The platform developed is to assist volunteers with time consuming and redundant tasks that are carried out on a weekly basis. It gives sports clubs a web presence and helps them move towards a more technology driven future. Members no longer need to go directly to club volunteers for information it will all be available on the platform. Payments no longer need to be handled by hand, as payments can be paid on the platform for memberships, pitch rentals and shop services.

Registrations for the platform is only required by club owners or volunteers. Other users such as members are not required to register as they can access any registered club without needing to sign in. When payments are being made for the previously mentioned services, un-registered users are required to fill out information to ensure clubs have the correct details.

## 1.2 Business Context

This online platform will allow sports clubs to sign-up for free to avail of the free services which include a club home page to share information and posts. Clubs will be able to share membership, team and pitch information at no extra cost. Ads relevant to the club will be displayed within their home page. (e.g. O'Neills adverts for GAA clubs).

Our premium services which include member registration, rostering/scheduling, rent-a-pitch and club shop must all be purchased either separately or as a discounted package.

Our goal is to help free up volunteers time, by reducing this work load it would prevent volunteers from potentially leaving clubs and would hopefully encourage more to join. The rent-a-pitch and club shop features open more opportunities for clubs to increase their revenue which could be used to improve facilities and equipment and hopefully encourage more members to join.

We believe our platform will accomplish all the goals mentioned above and we hope to continue to enhance current features and add more beneficial features to improve members and volunteers experience.

## 1.3 Glossary

**Python**
Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding allow for it to rapidly develop applications.

**Django**
Django is a Python-based free and open-source web framework, which follows the model-view-template architectural pattern. Which offers a wide variety of packages to overcome time consuming tasks.

**PyCharm**
PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCS), and supports web development with Django.

**SQLite**
SQLite is an in-process library that implements a self-contained, zero-configuration, serverless, transactional SQL database engine.

**HTML**
Hyper Text Markup Language, it is a language used to create content on the web, it defines structures and layouts using tags and attributes.

**CSS**
Cascading Style Sheets are used to describe the presentation of a document written in a mark-up language such as HTML.

**JavaScript**
JavaScript is a high-level, interpreted programming language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. Often used in basic HTML pages to enhance user interaction and overall attractiveness of the site.

**GitLab**
Is a web-based git-repository. GitLab allows us to collaborate on code, to help safely create and edit code on our project. It is free and open- source.

# 2. General Description

## 2.1 Product/System Function

Below is a combined list of registered clubs' admins, unregistered users and site admins functionality offered by our platform:

*Club Admins*

**Register**
Club owners/volunteers will have to register using first name, last name, email, username and password as these will be required to login. This is followed by club information which includes important details relating to their club such as name, address, contact details and club crest.

**Login / Logout**
Registered users will have access to login and logout to prevent security risks.

**Edit Profile Details**
Registered users will be able to edit profile information which includes; first name, last name, username, email and password.

**Contact Club-Kit**
Registered users have access to contact Club-Kit support staff using the available contact form implemented into the main site.

**Reset Password**
Registered users can reset passwords by sending verification to registered email when they have forgotten their password to log in.

**Edit Club Details**
Registered users can edit their club details when they require updating.

**View/Add/Delete Posts to/from Club Home Page**
Registered users can view, add or delete post which have been posted on their club home pages. Posts can include photos and text relating to events or information within their club.

**View/Add/Delete/Update Teams Details**
Registered users can view, add, delete or update team information within their club. These details are required for the roster feature as clubs require teams to amend slots to the roster.

**View/Add/Delete/Update Pitches Details**

Registered users can view, add, delete or update pitch information within their club. These details are required for both the roster and rent-a-pitch feature as teams are allocated slots on specific available pitches for the roster and users wishing to make a booking must select a pitch that is available to rent.

**View/Add/Delete/Update Membership Details**

Registered users can view, add, delete or update membership information within their club. These details are required for our membership registration feature as clubs may offer multiply membership options.

**View/Add/Update New Members**

Registered users will have access to view, add, delete and update member information within their club. This is important so club admins can handle their members information; such as contact details or incorrect inputted data when registering.

**View/Add/Delete/Update Roster Details**

Registered users can view, add, delete, or update roster details within their club. This is important so information being displayed is correct.

**View/Add/Delete/Update Rental Details**

Registered users can view, add, delete or update rental details within their club. This is important for cancellation, incorrect inputted details or bookings not made through the platform.

**View Shop**

Registered users can view their club shop from a user's perspective to ensure format and design is as expected. They will also be able to carry out all actions that a member can accomplish which include making purchases.

**View/Add/Delete/Update Shop Categories Details**

Registered user can view, add, delete or update the categories of products they offer within their club. This is important when adding products to their club shop, it allows us to categories shop items to make them easily filtered when being views by a user.

**View/Add/Delete/Update Shop Products Details**

Registered user can view, add, delete or update the products they offer within their club. This is important to ensure available products are visible to the end-user.  Products include a title, description, image, price, size and stock quantity.

**View Shop Orders**

Registered users will be able to view orders placed within the club.

*Club Members / Un-registered Users*

**Contact Club-Kit**

Un-registered users will have access to contact Club-Kit support staff using the available contact form implemented into the main site. Queries may include issues or bugs identified on the site.

**View Posts on Club Home Page**

Un-registered users will have access to view posts on a specific club home page.

**View Teams Details**

Un-registered users will have access to view team information associated to a specific club. Details containing member personal information will not be available.

**View Pitches Details**

Un-registered users will have access to view pitch information associated to a specific club. Details will include name of pitch, size, rental availability and pictures.

**View Membership Details**

Un-registered users will have access to view available membership information associated to a specific club. Details will assist the user with deciding to register with the club.

**Sign Up for a Club**

Clubs that pay for the registration feature allow non-members to sign-up for their club and make membership payments. Details will be available to club admins.

**View Roster Details**

Un-registered users will be able to view roster/schedules set by club admins. This information is displayed in a calendar.

**Rent Available Pitches**

Un-registered users have access to make bookings for available pitches which clubs offer for rental. Details of unavailable slots are displayed in a calendar.

**View Shop**

Un-registered users can view a list of all available products associated to a specific club.

**View Shop Products Details**

Un-registered users can select products they wish to see further details about.

**Filter Shop Categories**
Un-registered users can filter available products based on the category. Helpful for finding required products quickly.

**View/Add/Delete/Update Items in Cart**
Un-registered users can view, add, delete or update items within their cart. Allows for quick customization of their shopping cart.

**Checkout/Make Payment**
Un-registered users can check out once they've finished adding items to their cart. Payments are made using PayPal.

*Club-Kit Admins*

**Handle Data Within Admin Page**
Club-Kit admins will be able to access all registered clubs' information. This is restricted to superusers on this application.

## 2.2 User Characteristics and Objectives

This platform will be accessible online so it will be available by users with internet access. The platform renders sufficiently on both mobile and tablet devices. The intended use is for club admins, club members or simply users wishing to avail of club information and services. It acts as a webpage to display club details relating to contact details, notices, teams, pitches, memberships and roster/scheduling information. Whilst offering services to handle payments for registration, pitch rentals and club shop products. Club admins are responsible for updating club information, team and pitch details. These details are required to be updated to ensure additional features can be utilized. Clubs are also required to upload items to their shop if they wish to avail of this service.

## 2.3 Operational Scenarios

Before a club can get set up on our system, they will be required to register their personal and club details. The option to register will be displayed via our home page. Users wishing to access their clubs' pages do not require any sign in and a free to utilize the services without verification. If the club admin users wish to log in, the button is located to the right of our home page navigation bar. They will be required to enter a username and password. Once they've logged in the option to logout will be available in the same position as the log in was located.
When incorrect details are entered an error message will show on screen advising that either an invalid username or password has been entered. If user has forgotten their password, they can select forgotten password and they will be redirected to a change password page. Email verification will be required in order to change password.

Once a user successfully logs in and accesses their club through "Our Clubs" page located in the navigation bar of our main site. They will be redirected to their club's home page. Options to edit details of their club and add club posts will be available of this page.

There is two available tabs in the navigation bar to once a club admin is logged in to manage their team and pitch details. These are important aspects that clubs will need to include if their decide to use either our roster or rent-a-pitch services. If admins wish to update roster details, they will be required to select a team and pitch to add to the schedule. When a non-registered user wants to make a booking for a pitch slot, the selection of a pitch is required.

Clubs will be required to add membership types including details and prices in order to utilize our member registration feature. Non-members wish to register for a club they will be required to select the membership type so that correct membership details can be sent to the club admin.

Club admins will be required to set up their club shop, firstly they must add the category types of their items. Afterwards they can begin to add products to their shops including titles, category, descriptions, price, units, sizes, and pictures. Categories created will be selected when creating a product to allows for easy filtering of categories on their club pages.

Non-registered users can access club shops, add items to a cart, and check-out via PayPal. Orders will be sent to clubs to package and handle. Clubs will be responsible for distributing orders to customers.

## 2.4 Constraints

Below is a list of constraints that we may take into consideration to successfully complete our project.

**Time**

Meeting our completion deadline is the main constraint. Balancing assignments, studying for exams and other personal factors do impact the amount of time we've got to complete such a large project. At the beginning when we are asked to set out our Gantt chart to show where we expect to be at each stage of our project lifecycle. It had proven to be quite difficult as we had never undertaken such a big task. To provide a time to complete something we've never experienced developing is tough as we don't consider the time it takes to learn how to go about it.

**Knowledge/Skills**

Our experience is a big constraint to the development of our project. It's understandable as to how programmers with years of experience can give an estimate of time to complete a project or individual task. But for students still learning and trying new things it is difficult to give a specific time of completion as we're constantly researching, implementing, fixing, and testing. This really did impact our time as there had been occasions where we had spent a couple of weeks trying to figure out certain problems whereas now it would take us a couple of minutes to fix now. But we guess that's the whole purpose of this project is to gain these skills.

**Model Approach**

At the beginning of this project we believed adopting an agile approach to the development of this system was the way forward to the success of this project. Whilst it was somewhat beneficial as we were open to new ideas to make the system better, we quickly realised adding features to a project that has deadlines really put us under pressure. We noticed ourselves using more of a build and fix model which included a learning phase prior to our build. This shift in development approach was a constraint on our overall planning of the project.

**Testing**

Testing was another constraint during the development of our project. With little experience in designing unit tests we focused more on implementing all our specified features and conducting manual tests. But with the constant changes and fixes we had to carry out during the development of our project we did realise the importance of testing. This is something we would certainly consider designing as soon as possible for future projects.

## 3. Function Requirements

### 3.1 Main Website

**Description:**

This will be our official website. Detailing what we do, the packages we offer, pricing for our service, contact details to for potential customers to get in contact. We will display the clubs registered with us and provide links to access those club pages.

**Criticality:**

High – this is an important aspect to our business but does not require much functionality so time to develop is not a huge concern. Of course, it is the first thing potential customers will see so it is important to make a good first impression.

**Technical Issues:**

This raises no technical issues as it is just basic front-end development to portray our platform. This will be completed and populated once we've built the back-end functionality. Ensuring all hyperlinks are functional is the most important thing for this requirement.

**Dependencies:**

This function is not dependent on any other requirements.

### 3.2 Register

**Description:**

New clubs will be required to carry out this step-in order to sign their club up to Club-Kit. There will be a hyperlink on the home page of our site that will redirect users to a sign-up page where they will be required to provide personal details along with club information. We allow for any club to register an account for free, simply to avail of our club home page feature. They can display club information and add posts. Additional features do require purchasing.

**Criticality:**

High - This function is the most important aspect of the whole system. Without this we would not have any clubs signed up to our platform and in turn our users will have no club pages to access and avail of the services.

**Technical issues:**

All our forms are created in Django utilising the available libraries. The form will be designed using HTML and CSS. Widgets within the form will be created using bootstrap. Using Django, we will carry out validations on our forms. Username, email and club name checks will be conducted to validate if data already exists in our SQL database. A primary key will be assigned to new users and to the newly added club.

**Dependencies:**

This function requires our webpage to render correctly. To ensure users can navigate to the registration form.

## 3.3 User Login/Logout

**Description:**

User will have the ability to sign into their club pages to carry out changes which owners will only have access to do so. Once a user is finished, they can safely logout to avoid any unauthorised entry.

**Criticality:**

High – This is an important aspect which all systems that require authorization. For security reasons this is needed.

**Technical issues:**

The login form will be designed using HTML and CSS. Using Django, we can verify users' credentials in our SQLite database, if user credential are valid they will successfully log in. Otherwise error message will prompt advising username or password is invalid. Logout can simply be completed by selecting the logout button in the navbar, there is a Django function that carries out this request.

**Dependencies:**

This function is dependent on a user being registered on our platform. This feature requires the registration function to be working to ensure users can sign-up.

## 3.4 Edit Profile Details

**Description:**

This function will allow registered users to update their personal information. Option to change password is also available here. Username cannot be changed as it is already assigned a primary key. First name, last name, email and password can be changed.

**Criticality:**

Medium – The ability to update these details is beneficial but not necessarily critical. Whilst changing a password is important for security reasons. Option to reset password is available on login page.

**Technical issues:**

Form will be rendered on this page with already populated fields containing current details, user can change details. Form will be designed in HTML and CSS. When form is submitted Django will post updated details to SQLite database.

**Dependencies:**

User will need to be logged in, in order to carry out this function.

### 3.5 Contact Club-Kit

**Description:**

This function will allow both registered and non-registered users contact Club-Kit regarding queries and issues. This function can be accessed through the navigation bar on our main site page.

**Criticality:**

Medium – This is an important feature all businesses should provide to keep up customer service enquiries.

**Technical issues:**

This will be a basis contact form, users can provide their email, subject and description of enquiry. The form will be created using HTML and CSS. Emails will be sent to our Club-Kit email address.

**Dependencies:**

This function is not dependent on any other requirements.

### 3.6 Reset Password

**Description:**

This function will allow registered users to reset their passwords when forgotten. This can be accessed on the log in page. There is a hyperlink provided called forgot password this will redirect the user to reset their password. If invalid details are provided an error message will prompt advising.

**Criticality:**

High – This is an important feature as it will avoid users contacting Club-Kit to reset passwords manually.

**Technical issues:**

The form rendered on this page will already have populated fields containing current details, user can change details. Form will be designed in HTML and CSS. When form is submitted Django will post updated details to SQLite database.

**Dependencies:**

This function is not dependent on any other requirements. User must have an account to reset password.

## 3.7 Edit Club Details

**Description:**
This function allows registered users to update their clubs' details. All original details can be changed. This ensures correct details are provided to end users. Details include club name, contact number, email and address. This function can be accessed through the club's home page.
**Criticality:**
Medium – Although this is not critical to the system, it allows our system to keep details up to date in case of human error or newly obtained details.
**Technical issues:**
The form rendered on this page will already have populated fields containing current details, user can change details. Form will be designed in HTML and CSS. When form is submitted Django will post updated details to SQLite database.
**Dependencies:**
User will need to have a registered account with Club-Kit and be successfully logged in.

## 3.8 View/Add/Delete Posts

**Description:**
Registered users will have the ability to view, add and delete posts within their own club. This function can be accessed on the club home page. Non admin users only can view club posts. Clubs can share important information, scores and deals.
**Criticality:**
Low – This is not a critical component to our overall system. We felt this would be a nice touch to make the platform feel more like a website to our customers rather than just another system to carry out some functionality.
**Technical issues:**
The form is created using Django and designed in HTML and CSS. The form accepts a title, image and text. When form is submitted Django will post details to SQLite database. Upon selecting delete a JavaScript prompt box will display to confirm deletion.
**Dependencies:**
User will need to have a registered account with Club-Kit and be successfully logged in to add and delete posts. Alternatively, posts can be viewed with logging in.

## 3.9 View/Add/Delete/Update Teams Details

**Description:**

Registered users will have the ability to view, add, update and delete teams within their own club. This function can be accessed within the navigation bar within a club's page. Non admin users only can view teams associated to each club. Clubs can share the teams within their clubs.

**Criticality:**

High – This is an important function for our platform as the information provided here is required for other functions available.

**Technical issues:**

The form is created using Django and designed in HTML and CSS. A post request is used to add details to our SQLite database. When editing a team, an auto populated form will be provided containing details of selected team and the registered user can change details then post update to the database. Upon selecting delete a JavaScript prompt box will display to confirm deletion.

**Dependencies:**

This feature does depend on the user being registered and is required for our roster feature. Un-registered users can freely view teams associated to a club.

### 3.10 View/Add/Delete/Update Pitches Details

**Description:**

Registered users will have the ability to view, add, update and delete pitches within their own club. This function can be accessed within the navigation bar within a club's page. Non admin users only can view pitches associated to each club. Clubs can share the pitches within their clubs.

**Criticality:**

High – This is an important function for our platform as the information provided here is required for other functions available.

**Technical issues:**

The form is created using Django and designed in HTML and CSS. A post request is used to add details to our SQLite database. When editing a pitch, an auto populated form will be provided containing details of selected pitch and the registered user can change details then post update to the database. Upon selecting delete a JavaScript prompt box will display to confirm deletion.

**Dependencies:**

This feature does depend on the user being registered and is required for our roster and rent-a-pitch features. Un-registered users can freely view pitches associated to a club.

### 3.11 View/Add/Delete/Update Membership Details

**Description:**

Registered users will have the ability to view, add, update and delete memberships within their own club. This function can be accessed within the navigation bar within a club's page. Non admin users only can view available memberships associated to each club. Clubs can share the memberships available to members within their clubs.

**Criticality:**

High – This is an important function for our platform as the information provided here is required for other functions available.

**Technical issues:**

The form is created using Django and designed in HTML and CSS. A post request is used to add details to our SQLite database. When editing a membership, an auto populated form will be provided containing details of selected membership and the registered user can change details then post update to the database. Upon selecting delete a JavaScript prompt box will display to confirm deletion.

**Dependencies:**

This feature does depend on the user being registered and is required for our membership feature. Un-registered users can freely view memberships associated to a club.

### 3.12 View/Add/Update Members

**Description:**

Registered users will have the ability to view, add and update member details within their own club. This function can be accessed within the navigation bar within a club's page. Non admin users cannot access this page.

**Criticality:**

Medium – Although this is not critical to the system, it allows our clubs to keep members details up to date in case of human error or newly obtained details.

**Technical issues:**

The form is created using Django and designed in HTML and CSS. A post request is used to add new members to our SQLite database. When editing a member's details, an auto populated form will be provided containing details of selected member and the registered user can change details then post update to the database.

**Dependencies:**

This function is not dependent on any other requirements. User must have an account to change details regarding members.

### 3.13 View/Add/Delete/Update Roster Details

**Description:**

Registered users will have the ability to view, add, update and delete roster information within their own club. This function can be accessed within the navigation bar within a club's page. Upon adding slots to rosters clubs can specify re-occurring events to save time for future entries. Non admin users can view these details but cannot modify or add details in any way.

**Criticality:**

High – This is one of our main features incorporated into our platform. It is one of the paid services we offer to clubs.

**Technical issues:**

The form is created using Django and designed in HTML and CSS. A post request is used to add roster information to our SQLite database. When editing roster details, an auto populated form will be provided containing details of selected roster slot and the registered user can change details then post update to the database. The roster details are displayed using a JavaScript calendar. Form will render team and pitch options for admins to select from.

**Dependencies:**

This function requires for a user to be logged into their club. A club must contain at least one team and one pitch in order to add roster information to their club schedule. These can be adding in previous mentioned functionality.

## 3.14 View/Add/Delete/Update Rental Details

**Description:**

Registered users will have the ability to view, add, update and delete rental information within their own club. This function can be accessed within the navigation bar within a club's page. Adding rental information should only be necessary when clubs take booking through an alternative source, but we hope our service will encourage clubs to redirect customers to the platform to book. Booked slots will display as unavailable to non-registered users but club admins can view details.

**Criticality:**

High – This is one of our main features incorporated into our platform. It is one of the paid services we offer to clubs.

**Technical issues:**

The form is created using Django and designed in HTML and CSS. A post request is used to add rental information to our SQLite database. Payment will be required when a non-registered user wishes to make a booking. They will be redirected to PayPal to complete booking. Club admins will have the ability to edit rental details, an auto populated form will be provided containing details of selected rental details and the registered user can change details then post update to the database. The roster details are displayed using a JavaScript calendar. Form will render the club's available pitches for rental to allow for user selection.

**Dependencies:**

Non-registered users can access this page, but clubs' admins must ensure pitches have been added to their club for customers to make bookings. Pitches must be set to available to rent for them to be show in the form.

## 3.15 Sign Up for a Club

**Description:**

This function will allow non-registered users to sign up as members for a club. They fill out a generic registration form, select membership type and process payment. Clubs will be responsible for adding available memberships for user to choose from.

**Criticality:**

High – This is one of our main features incorporated into our platform. It is one of the paid services we offer to clubs. We expect this to be the package most clubs will avail of.

**Technical issues:**

The form is created using Django and designed in HTML and CSS. Once form is complete users can make a payment. Once this is successful club admins can view paid members. Payments will be handled using PayPal.

**Dependencies:**

This is dependent on the club admins adding memberships to club for users to choose from.

### 3.16 View/Add/Delete/Update Shop Categories Details

**Description:**

Registered users will have the ability to view, add, update and delete categories associated to the products available to purchase in their club shop. Clubs will need to add categories prior to adding products to their shop. It allows for better organization of products and prevents human error. Drop down menu is provided containing added categories when inserting new products to club shop.

**Criticality:**

High – This is associated to one of our main features incorporated into our platform. It is one of the paid services we offer to clubs.

**Technical issues:**

The form is created using Django and designed in HTML and CSS. A post request is used to add categories types to our SQLite database. Club admins will have the ability to edit category details, an auto populated form will be provided containing details of selected category and the registered user can change details then post update to the database.

**Dependencies:**

Non-registered users can't access this page, clubs' admins must ensure categories are created before adding products to store.

### 3.17 View/Add/Delete/Update Shop Products Details

**Description:**

Registered users will have the ability to view, add, update and delete products in their club shop. Clubs will provide a title, description, image, price, size and stock quantity. These products will be available for customers to purchase.

**Criticality:**

High – This is associated to one of our main features incorporated into our platform. It is one of the paid services we offer to clubs.

**Technical issues:**

The form is created using Django and designed in HTML and CSS. A post request is used to add products to our SQLite database. Club admins will have the ability to edit product details, an auto populated form will be provided containing details of selected product and the registered user can change details then post update to the database. Club Id must be associated with each product to ensure correct items are displayed to the end customer.

**Dependencies:**

Non-registered users can't access this page. Club admins cannot add products to their club without adding categories to their club.

### 3.18 View Shop

**Description:**

Non-registered users will have the ability to view all products available to purchase from any club. Product titles, images and price will be seen on this page.

**Criticality:**

High – This is critical to the system as it is one of our main packages that require purchasing. This allows clubs to share their products with their end customers which will then be available to purchase.

**Technical issues:**

The structure of the products on this page is designed using HTML and CSS. The data is rendered on this page using Django. Hyperlinks are available under each product to redirect to a product details page where then can then be added to a cart.

**Dependencies:**

This depends on clubs populating their stores with products. Without these being added there would be no products visible to the end user to purchase.

### 3.19 Filter Shop Categories

**Description:**

This function is to improve the end-users experience. Implementing a filter option will allow end-users to easily search for their desired products.

**Criticality:**

Low – This function is not critical to the overall system. We felt it would enhance the end-users experience and make the shop look more profession.

**Technical issues:**

Rending the available categories associated to a club using Django we could show this filter option within each club shop main page. Category list was designed using HTML and CSS.

**Dependencies:**

This list will only be displayed if there are categories associated to a club. Products will only display if they have been added by club admins.

### 3.20 View Shop Products Details

**Description:**

Non-registered users will be able to click on product within a club shop. This function will redirect them to the product detail page where they can see further details about a product. Select a size and add the product to a cart.

**Criticality:**

High – This is an important function to the completion for making a purchase of club products. Without this it would not be possible to complete an order.

**Technical issues:**

This page is rendered using HTML, CSS, JavaScript and Django. The add to cart button sends a POST request and adds the item to a cart session.

**Dependencies:**

This is depended on products being available to the end user. This function is not available if there are no products visible.

### 3.21 View/Add/Delete/Update Items in Cart

**Description:**

Non-registered users will have been able to view, delete and update products in their cart. Products can be added through the product details page. Details regarding cart information will be displayed to the end-user to ensure they have the correct products.

**Criticality:**

High – It is important to show end-users a list of products they wish to purchase prior to making payment. This is standard with most e-commerce businesses. Once end-users are happy to proceed, they will be redirected to a check out page.

**Technical issues:**

The structure of the cart is designed using HTML and CSS. Utilizing a session key for the cart we could ensure modifications to the cart would update as required.

**Dependencies:**

This is dependent on products being adding to the cart by the end user. Although this page is visible when no items added it will show as empty to the end-user.

### 3.22 Checkout/Make Payment

**Description:**

When non-registered users are content with the products in their cart then can check-out. They will be redirected to an order form which will require personal details after this is complete, they will be redirected to PayPal to complete payments. Once complete order details will be available to clubs which the purchase is being made to.

**Criticality:**

High – This is the final stage required in making a purchase within a club. This function is one of our main features and is critical to the overall project requirements.

**Technical issues:**

The form and PayPal payment process is developed using Django. The form is designed using HTML and CSS. POST requests upon completion send order form details to our SQLite database which can then be viewed by club admins.

**Dependencies:**

This is dependent on products being added to a cart prior to check-out.

### 3.23 View Shop Orders

**Description:**

Club admins will have the ability to view shop orders associated with their clubs. This is needed so admins can handle and distribute orders to their customers / members.

**Criticality:**

High – This is an essential function to the shop feature. Clubs need to know of orders placed and the details to ensure they can successfully deliver products to the end-users.

**Technical issues:**

The details of order will be rendered using Django. Structure of this page will be designed using HTML and CSS.

**Dependencies:**

This page is not dependent on any other function, but details will only be visible if orders are successfully placed within a club.

### 3.24 Handle Data Within Admin Page

**Description:**

Club-Kit admins will have full access to view and modify all data within the Club-Kit platform.

**Criticality:**

High – This offers a better interface to view data within our Django application rather than using SQLite database.

**Technical issues:**

This is an implemented feature of Django and requires no technical issues.

**Dependencies:**

This is not dependent on any other function within our system.

## 4. System Architecture

Our system architecture is a typical 3 layered structure comprising of a client – server – database. Club-Kit website and individual club pages will be our client side or front-end. This will be developed using HTML, CSS and JavaScript. End-users will interact with the application and send requests to our application server layer.

Requests and application logic in our server layer is written in Django. This will provide the end-user with requested services. We utilized get, post and delete methods to handle these requests. Our application layer inserts/retrieves the requested data to our database layer.

Validations occur in our application layer to ensure valid data is being inputted into our SQLite database. Once we decide to deploy our application, we will migrate to MySQL database as our SQLite database is only beneficial for development purposes.

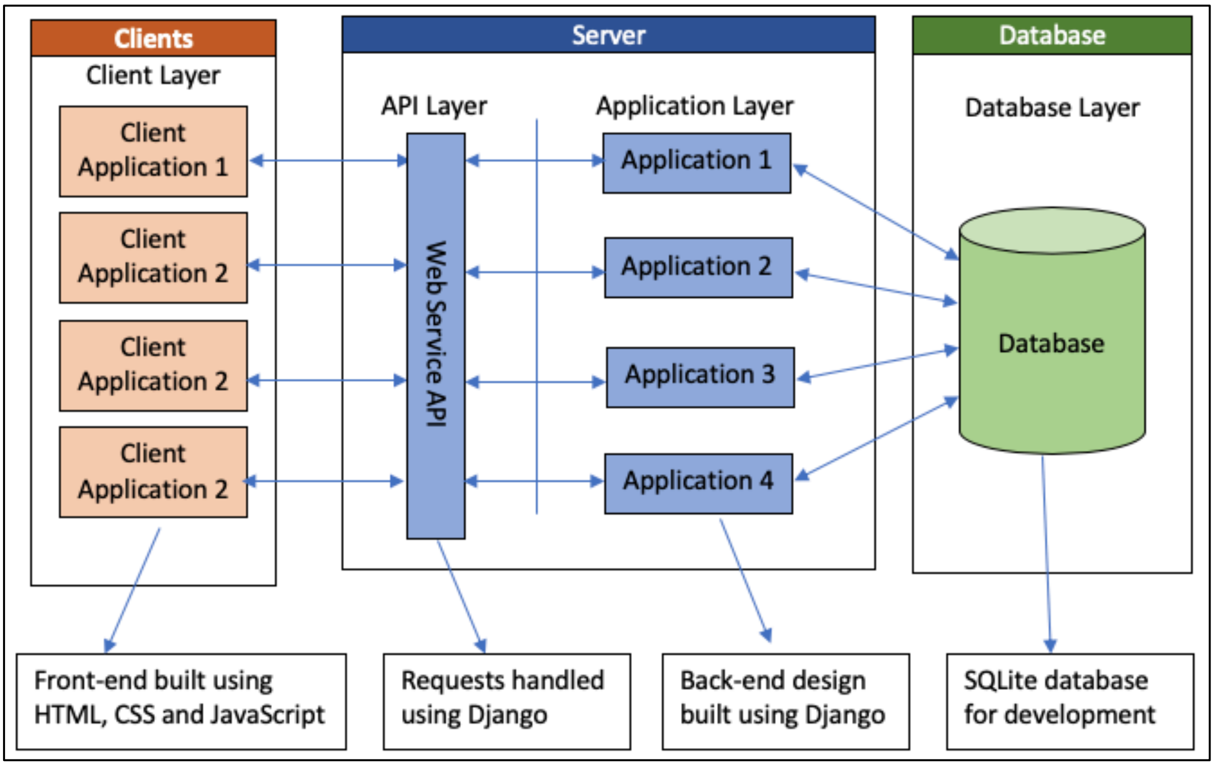


**Fig 4.1**

## 5. Future Development

### 5.1 Future Modifications to Current Features

- Cater for all types of sports clubs (different processes and activities).
- Wider variety of membership registration forms.
- Multiple club admins.
- Customizable club home page.
- Member login (avail of other services that club's want to share with members).
- Comments on club posts.
- Alternative payment options (Currently PayPal).
- Contact form on club home page.
- Map of club location on club home page.
- Implement multiple photos option (shop products).
- Reports (members, rosters, pitch bookings and shop orders).
- Set up notifications for roster details.
- Implement logging of requests and actions on all features.
- Expand team feature to include player details (reports on players).
- Allow clubs to assign members to teams.
- Renewal of club features system.
- Handle returns for shop orders.
- Implement refund process for cancelations.

### 5.2 Future Development for New Features

- Mobile application (currently mobile compatible)
- Built APIs to allow clubs to incorporate features on their own websites.
- Games and competition management tool.
- Manager platform to see available players and sort teams for upcoming events.
- Referee and official's management tool.
- Handle payments for upcoming fundraisers, games or events.
- Offer live scoring.
- Fixtures and future events.
- League tables.
- Communication system for teams and managers to communicate on.
- Reminder system for managers, teams or all members.

### 5.3 Future Tools to Manage System

- ELK stack to handle logging.
- Migrate from SQLite database to MySQL.
- Google App Engine to handle infrastructure.

- Utilize google app engines cron jobs and task queues to automate tasks. (e.g. Sending email reminder for renewal of memberships and club package subscriptions).

# 6. High Level Design

## 6.1 Club Admin Set Up Functionality

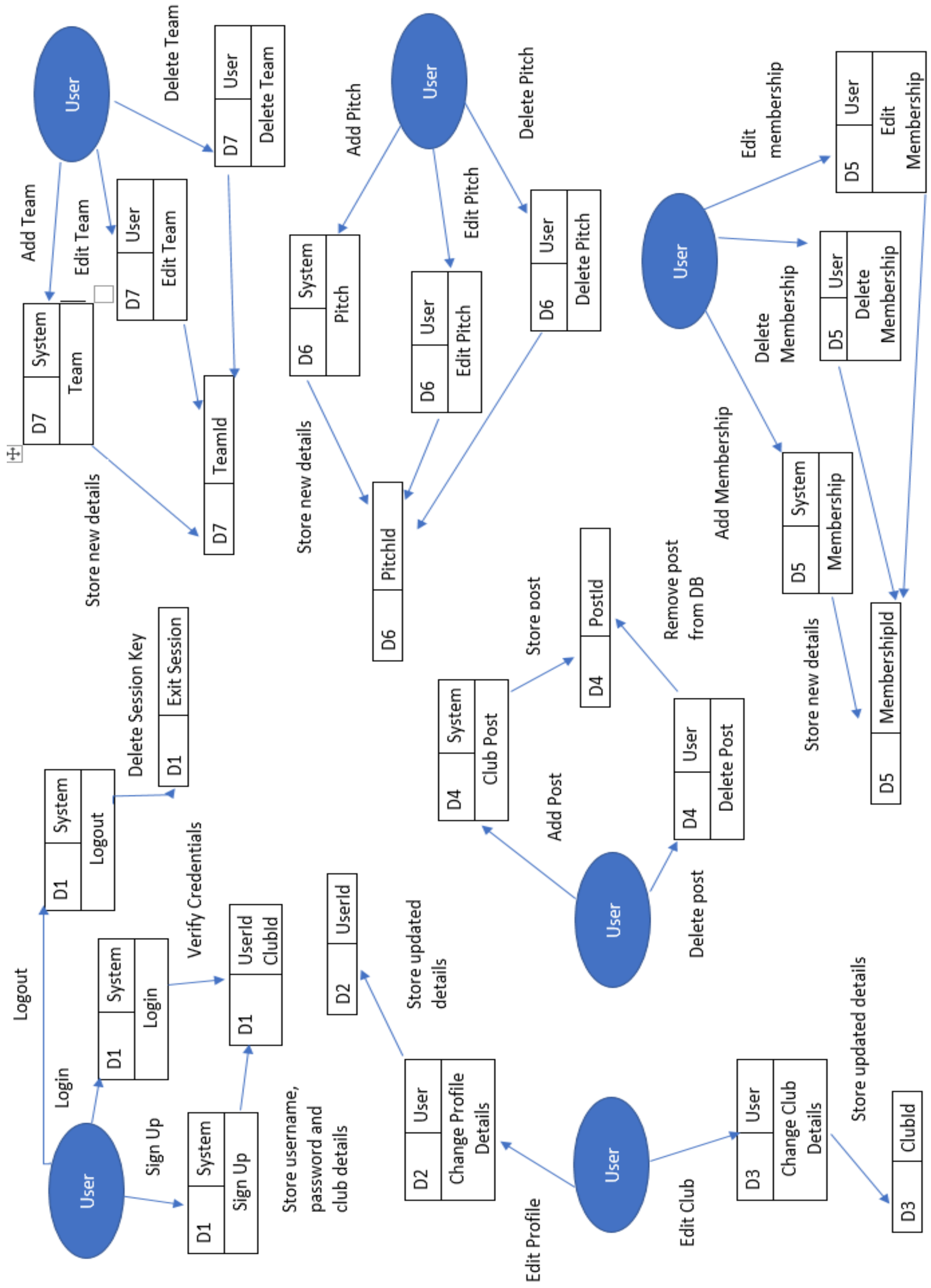


**Fig 6.1**

### 6.2 Club Admin Paid Services Functionality

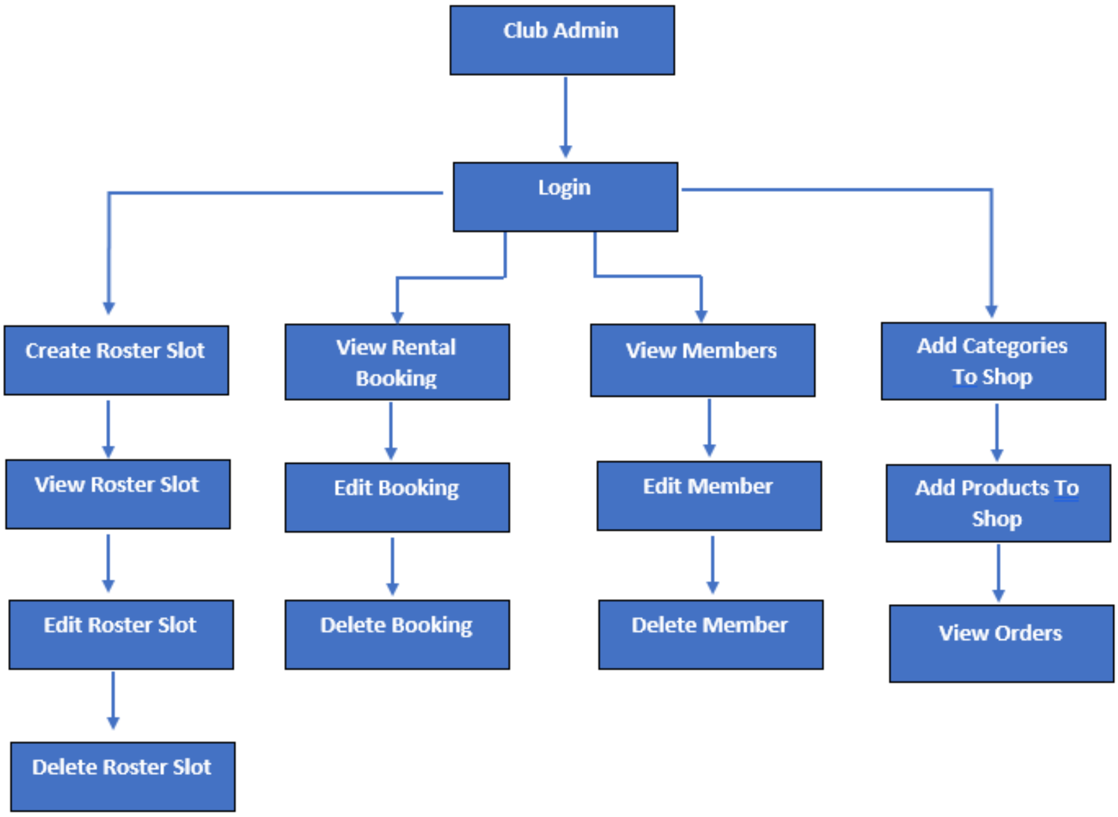


Fig 6.2

### 6.3 User Paid Services Functionality

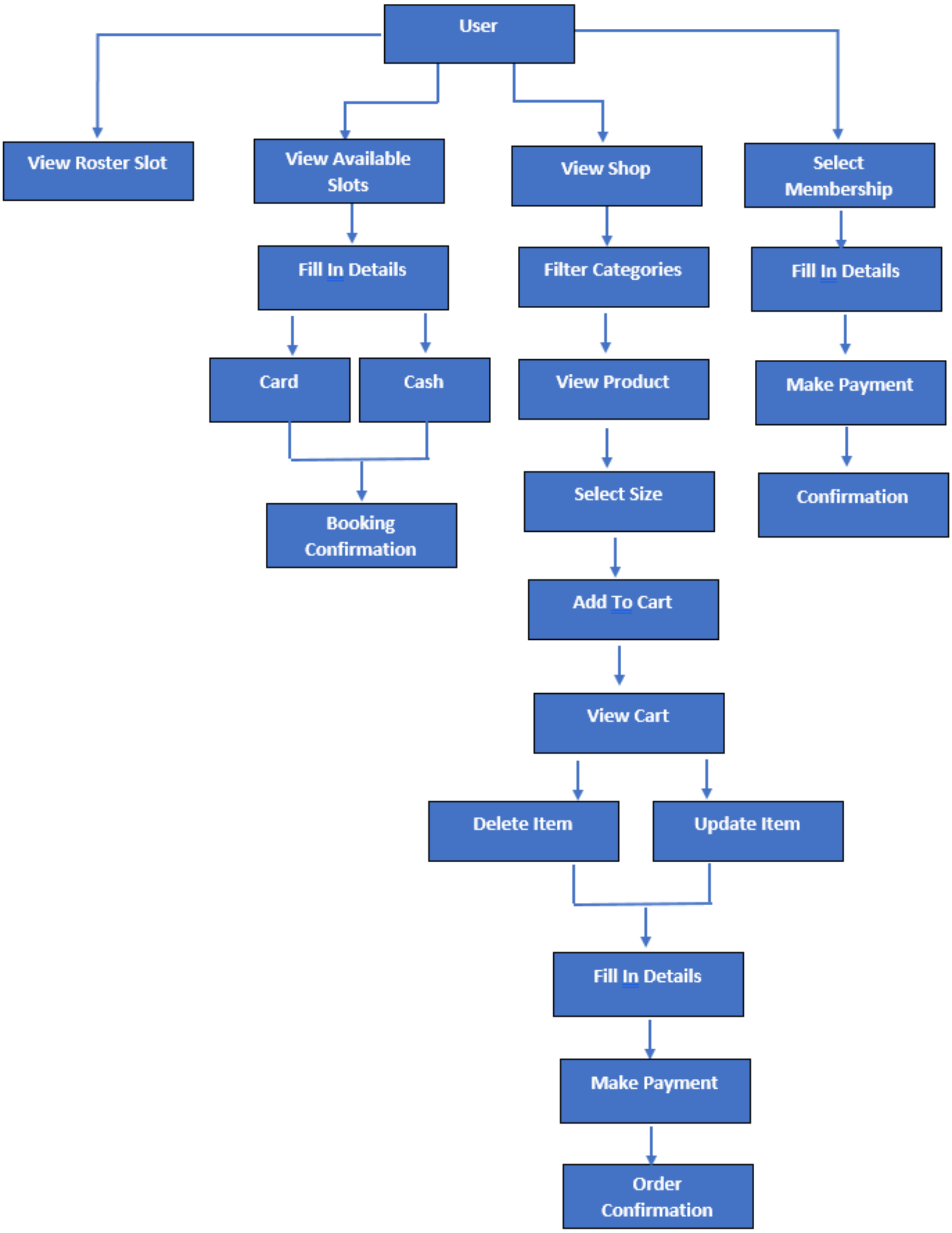
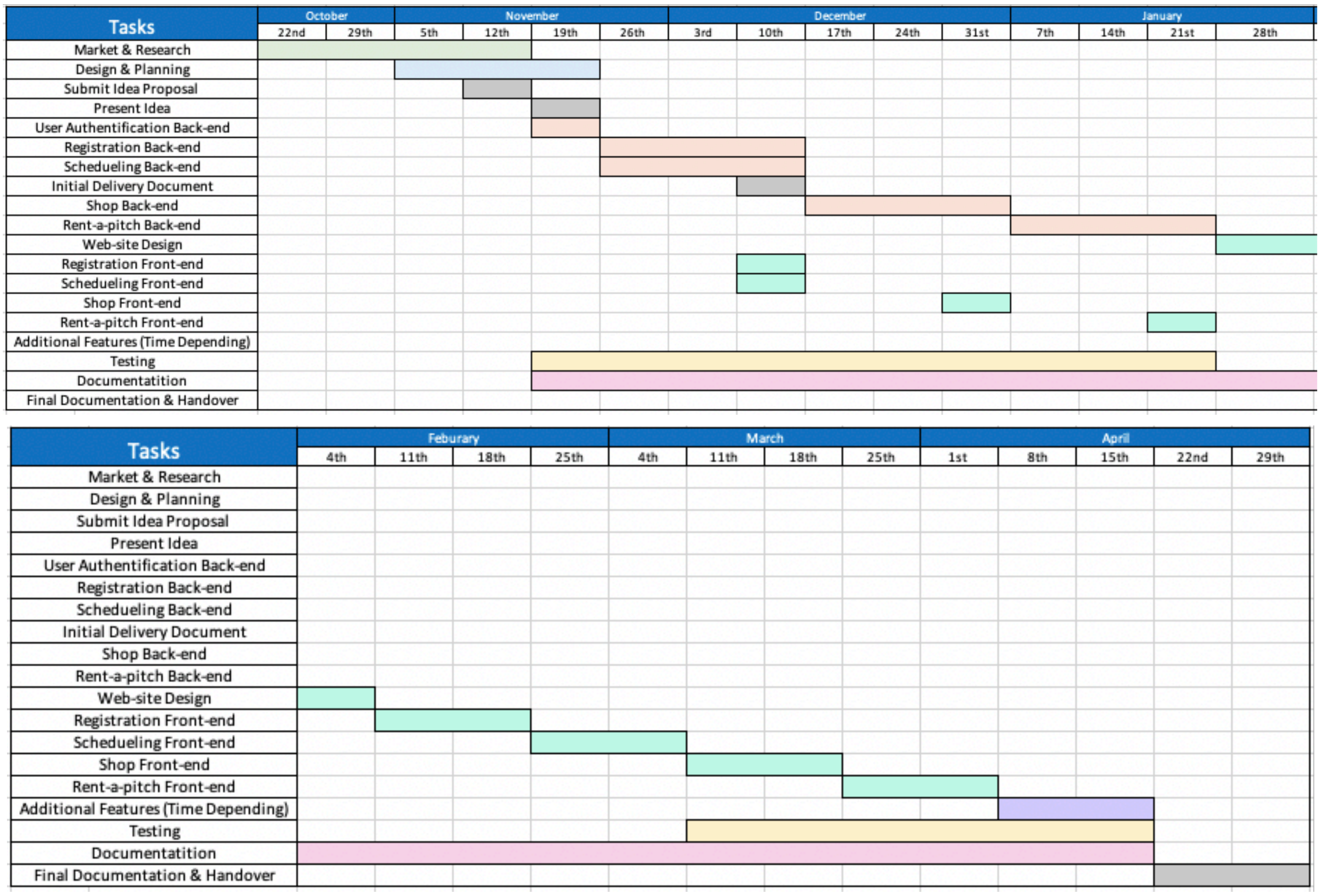


Fig 6.3

# 7. Preliminary Design

## 7.1 Gantt Chart



| Tasks | October | | November | | | | December | | | | | January | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 22nd | 29th | 5th | 12th | 19th | 26th | 3rd | 10th | 17th | 24th | 31st | 7th | 14th | 21st | 28th |
| Market & Research | ■ | ■ | ■ | ■ | | | | | | | | | | | |
| Design & Planning | | | ■ | ■ | ■ | | | | | | | | | | |
| Submit Idea Proposal | | | | ■ | | | | | | | | | | | |
| Present Idea | | | | | ■ | | | | | | | | | | |
| User Authentification Back-end | | | | | ■ | | | | | | | | | | |
| Registration Back-end | | | | | | ■ | ■ | ■ | | | | | | | |
| Schedueling Back-end | | | | | | ■ | ■ | ■ | | | | | | | |
| Initial Delivery Document | | | | | | | | ■ | | | | | | | |
| Shop Back-end | | | | | | | | | ■ | ■ | ■ | | | | |
| Rent-a-pitch Back-end | | | | | | | | | | | | ■ | ■ | ■ | |
| Web-site Design | | | | | | | | | | | | | | | ■ |
| Registration Front-end | | | | | | | | ■ | | | | | | | |
| Schedueling Front-end | | | | | | | | ■ | | | | | | | |
| Shop Front-end | | | | | | | | | | | ■ | | | | |
| Rent-a-pitch Front-end | | | | | | | | | | | | | | ■ | |
| Additional Features (Time Depending) | | | | | | | | | | | | | | | |
| Testing | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | |
| Documentatition | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| Final Documentation & Handover | | | | | | | | | | | | | | | |

| Tasks | Feburary | | | | March | | | | April | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 4th | 11th | 18th | 25th | 4th | 11th | 18th | 25th | 1st | 8th | 15th | 22nd | 29th |
| Market & Research | | | | | | | | | | | | | |
| Design & Planning | | | | | | | | | | | | | |
| Submit Idea Proposal | | | | | | | | | | | | | |
| Present Idea | | | | | | | | | | | | | |
| User Authentification Back-end | | | | | | | | | | | | | |
| Registration Back-end | | | | | | | | | | | | | |
| Schedueling Back-end | | | | | | | | | | | | | |
| Initial Delivery Document | | | | | | | | | | | | | |
| Shop Back-end | | | | | | | | | | | | | |
| Rent-a-pitch Back-end | | | | | | | | | | | | | |
| Web-site Design | ■ | | | | | | | | | | | | |
| Registration Front-end | | ■ | ■ | | | | | | | | | | |
| Schedueling Front-end | | | | ■ | ■ | | | | | | | | |
| Shop Front-end | | | | | | ■ | ■ | | | | | | |
| Rent-a-pitch Front-end | | | | | | | | ■ | ■ | | | | |
| Additional Features (Time Depending) | | | | | | | | | | ■ | ■ | | |
| Testing | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | | |
| Documentatition | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | | |
| Final Documentation & Handover | | | | | | | | | | | | ■ | ■ |

Fig 7.1

# 8. Appendices

## 8.1 Resources

Research and quick answers - www.google.com
Django tutorials by Max Goodridge -
https://www.youtube.com/channel/UCAx4nmhI7S1RcPiaG-Uw0tg
Django tutorials by Coding Entrepreneurs - https://www.youtube.com/user/CodingEntrepreneurs
Django tutorials by Clever programmer -
https://www.youtube.com/channel/UCqrILQNl5Ed9Dz6CGMyvMTQ
Django tutorials by Traversy Media - https://www.youtube.com/user/TechGuyWeb
Django tutorials by the Dumbfounds (testing)-
https://www.youtube.com/channel/UC33uwXXDrI5TxG4IXnjS28g
Django tutorials by Mike Hibbert (sessions and payments) -
https://www.youtube.com/user/MickeySoFine1972
Django support - https://docs.djangoproject.com/en/2.2/
Django tutorials for testing - https://realpython.com/
Django tutorials for URLs, models and views - https://www.tutorialspoint.com/
Paid course to enhance knowledge for building back-end https://www.udemy.com/full-stack-development-web-app-mobile-app-back-end-api/
Shop Tutorial - https://www.packtpub.com/application-development/django-example-video#
JavaScript calendar for Roster and Rent-a-pitch - https://fullcalendar.io
Hover.css effects - https://ianlunn.github.io/Hover/
Animate.css effects - https://daneden.github.io/animate.css/
FontAwesome icons - https://fontawesome.com
Bootstrap 4 - https://getbootstrap.com

# 9. UML Use Cases

## 9.1 Club Home Page

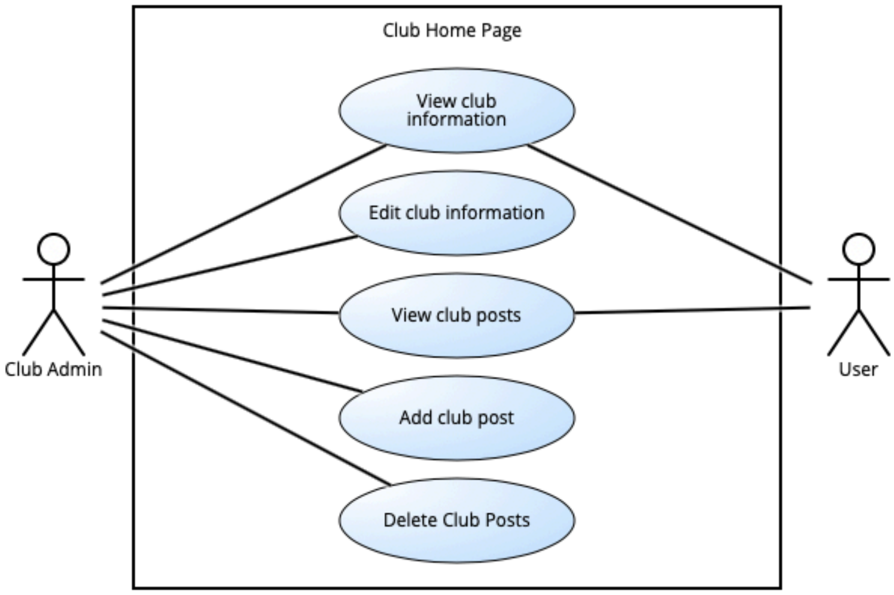


**Fig 9.1**

## 9.2 Club Team / Pitches

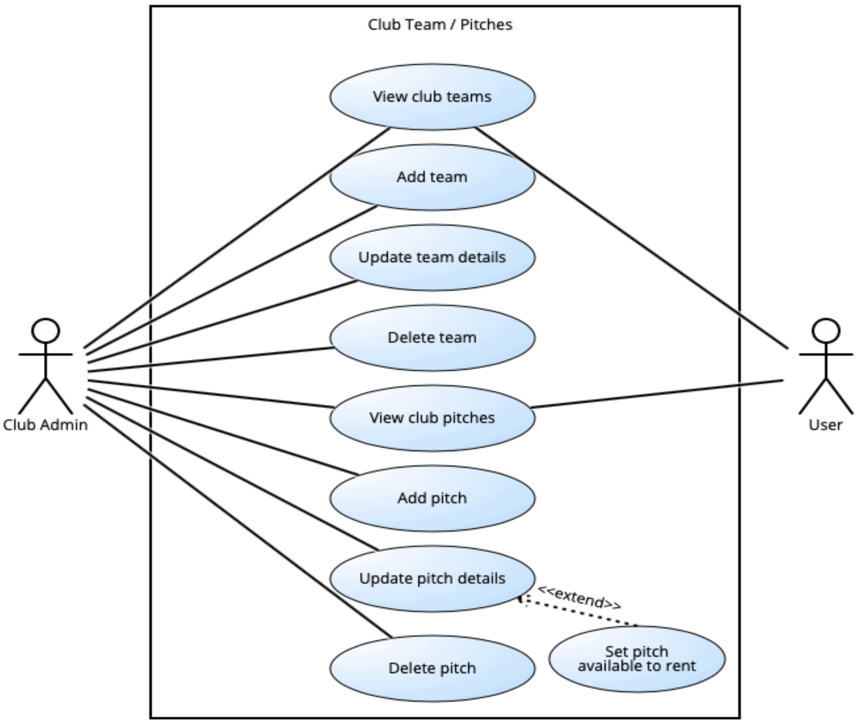


**Fig 9.2**

## 9.3 Member Registration

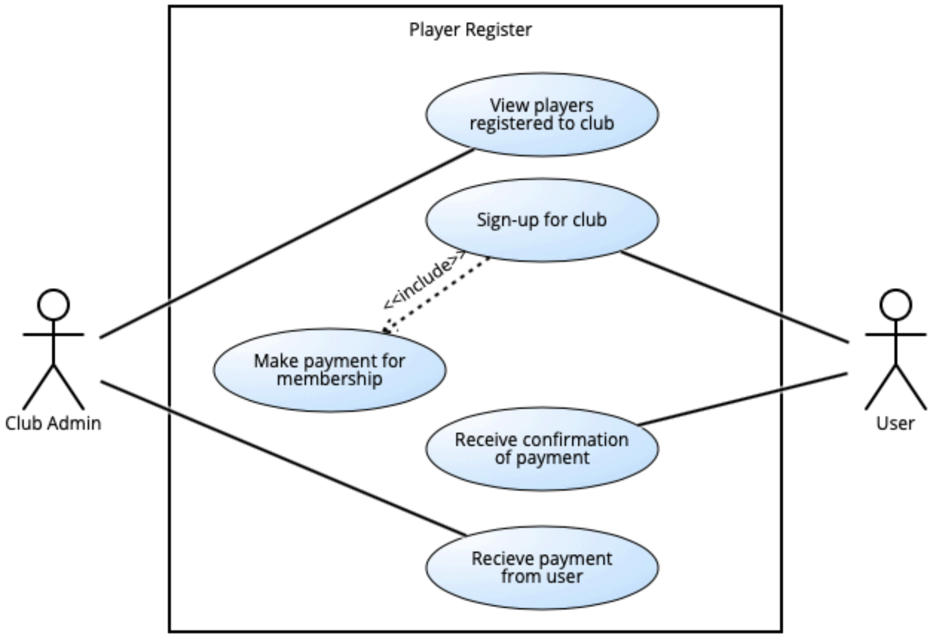

Player Register
View players registered to club
Sign-up for club
<<include>>
Make payment for membership
Receive confirmation of payment
Recieve payment from user
Club Admin
User

Fig 9.3

## 9.4 Club Roster / Rent-a-Pitch

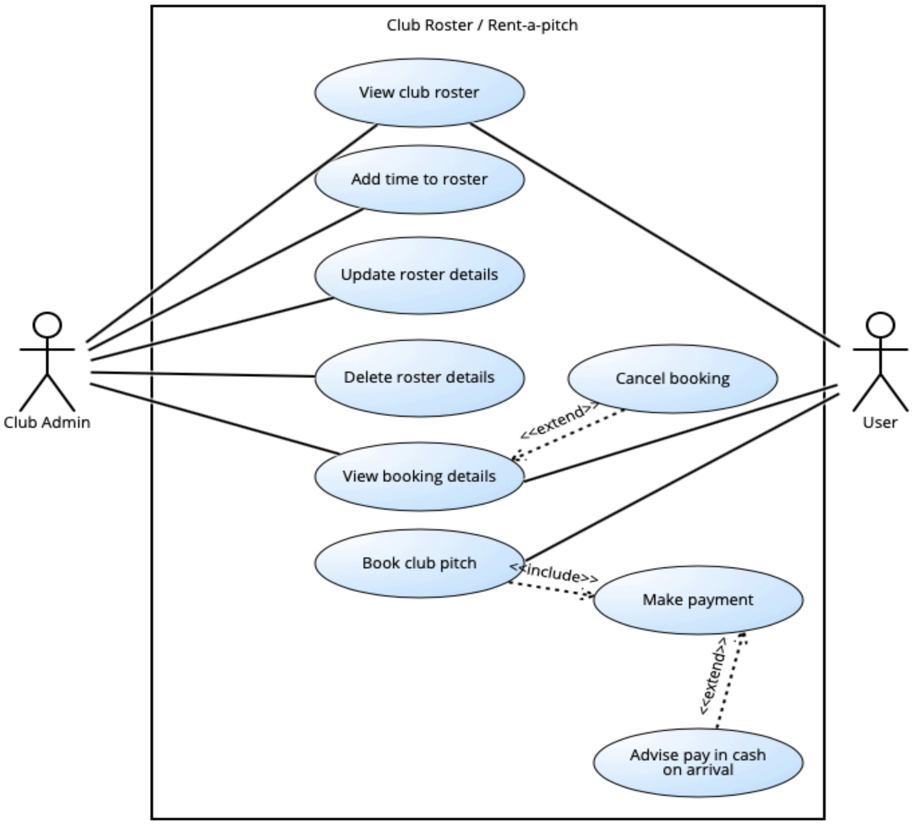


Fig 9.4

### 9.5 Club Shop

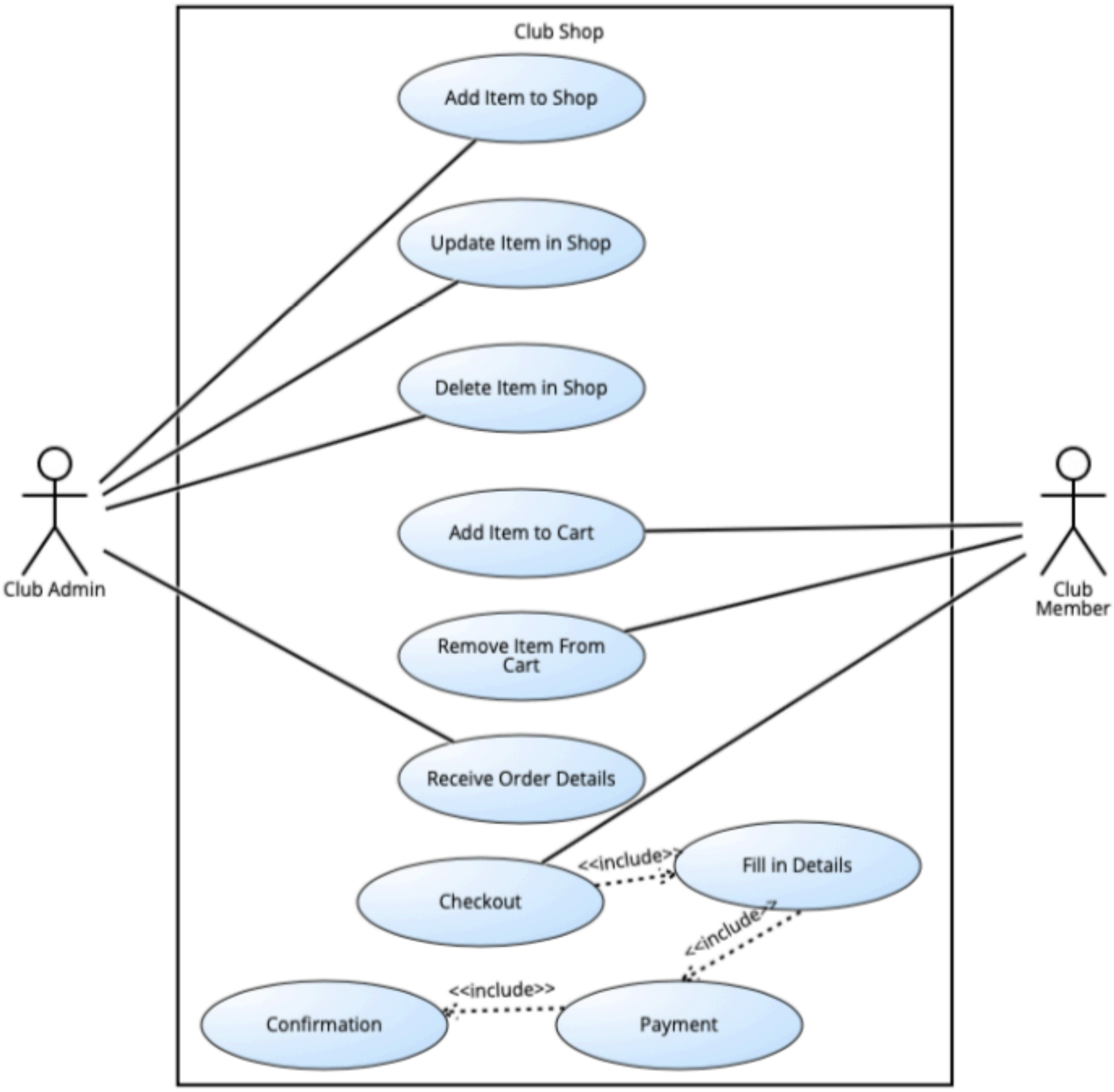


Fig 9.5

## 10. Database Structure & Table Relationships

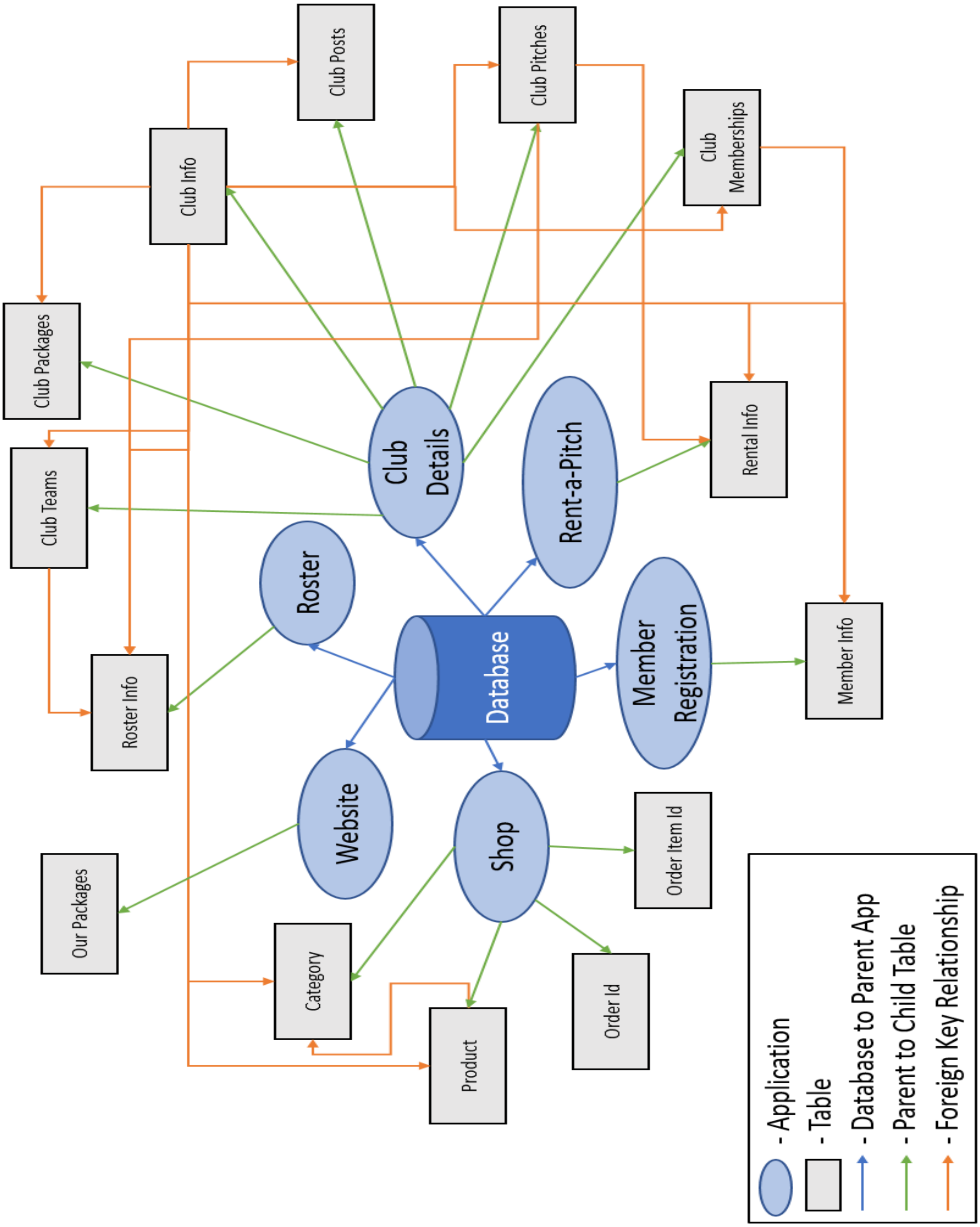


Fig 10.1

## 11. Technologies Used

**Application Platform:** Our application will be built as a web-app, but we will modify the front-end to be mobile friendly. We believe there is no need to develop a mobile application for the prototype as the main functionality is for admin work which is not suitable to be carried out on a smart phone, but it is necessary for it to be compatible on smart phones for our features such as pitch rental and club shops. In the future we would look at developing a mobile application.

**IDE (Integrated Development Environment):** For our IDE we chose PyCharm due to it being highly recognised within the python community and built specifically for python.

**Database:** For the development stages of our application we used SQLite database. But once we are at the stage to push to production, we will migrate to a MySQL is the database as it is faster, more reliable and cheaper than other database options out there. There is also a lot of documentation online to support our database needs. MySQL works well with many programming languages which is important and includes all the necessary features we need to develop our application system.

**Git Repository:** GitLab is our web-based git-repository of choice. GitLab allows us to collaborate on code, to help safely create and edit code on our project. It is free and opensource.

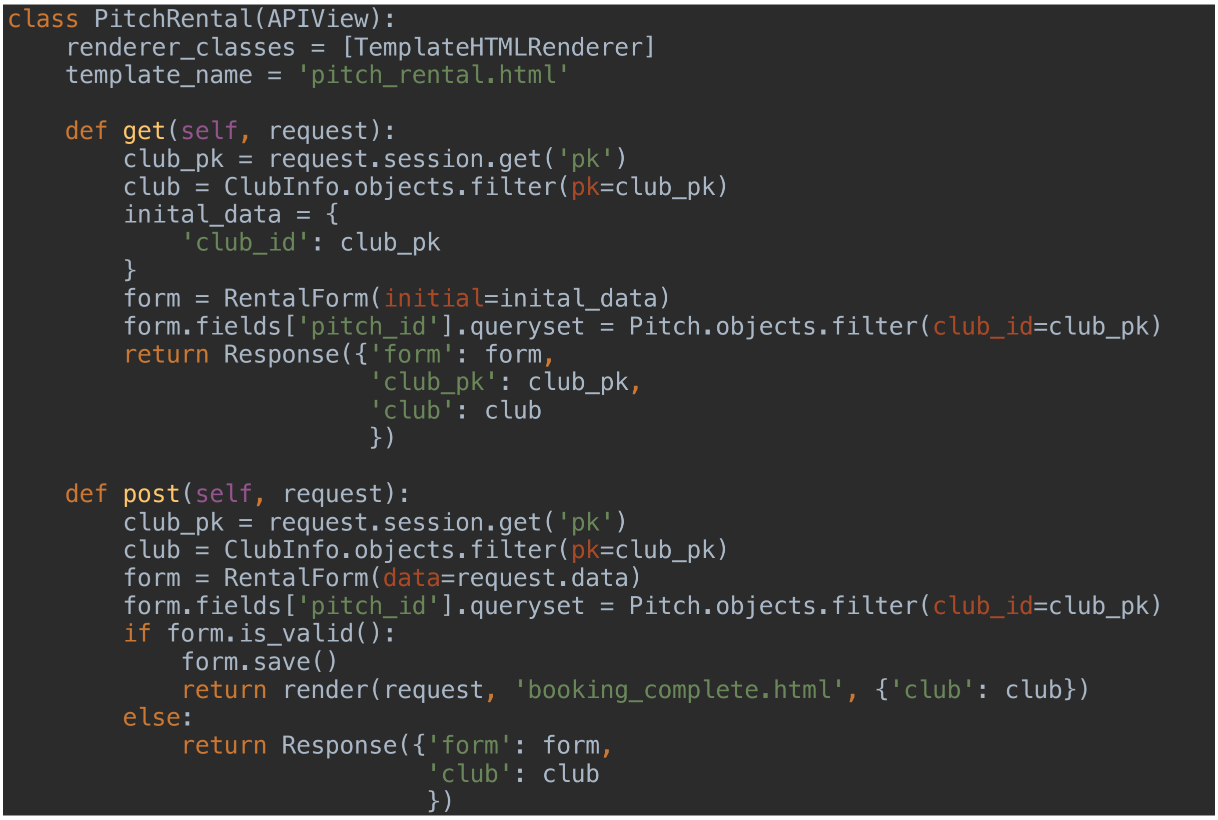
**Back-End:** Python is our preferred object orientated programming language which we have experiencing using. Using python, we will build our back-end functionality and API's. Using Django, with prior experience using this framework we decided to try and learn something new. We touched on Django Rest Framework in our development of the back-end, but did not use it thought-out the design as we had more experience using Django and felt more comfortable using this framework.

**Front-End:** We had initially hope to build our front-end using Angular unaware of the high learning curve associated with it. Due to this we've decided to develop our front end using basic Html, CSS, and JavaScript with no framework to support these. It is our goal to learn this in the future but we just felt tackling this required learning will slow down our developing process for our outlined features.

## 12. Source Code Highlights

```
<script>
$(function() {

// page is now ready, initialize the calendar...

$('#calendar').fullCalendar({
    height: 650,
    selectable: true,
    minTime: '09:00:00',
    defaultView: 'agendaFourDay',
    groupByDateAndResource: true,
    header: {
      left: 'prev,next',
      center: 'title',
      right: 'agendaDay,agendaFourDay'
    },
    views: {
      agendaFourDay: {
        type: 'agenda',
        duration: { days: 5 }
      }
    },
    eventRender: function(eventObj, $el) {
      $el.popover({
        title: eventObj.title,
        content: eventObj.description,
        trigger: 'hover',
        placement: 'top',
        container: 'body'
      });
    },

    events: [ // put the array in the `events` property

        {% for info in bookings %}
                {
                {% if user.is_authenticated %}

                    title  : '{{ info.name }}',
                    description : '{{ info.name }} {{ info.pitch_id }} {{
info.email }} {{ info.mobile }}',
                    start  : '{{ info.date|date:"Y-m-d" }}T{{
info.start_time|date:"G:i:s" }}',
                    end  : '{{ info.date|date:"Y-m-d" }}T{{
info.finish_time|date:"G:i:s" }}',
                {% else %}
                    title  : 'BOOKED',
                    description : 'There is already a booking in this slot',
                    start  : '{{ info.date|date:"Y-m-d" }}T{{
info.start_time|date:"G:i:s" }}',
                    end  : '{{ info.date|date:"Y-m-d" }}T{{
info.finish_time|date:"G:i:s" }}',
                {% endif %}
                },
        {% endfor %}

      ],

  });

});
</script>
```

Above is the code used to render the calendar for our rent-a-pitch feature. It's an open source calendar available from FullCalendar. It has been modified to suit our application. It will pull in bookings from the database and display them on the calendar. It displays, who booked the pitch, when and various contact details for the person. If the user is not authenticated, meaning they aren't an admin for the club, then the calendar only displays 'BOOKED' and does not reveal any details of the booking.



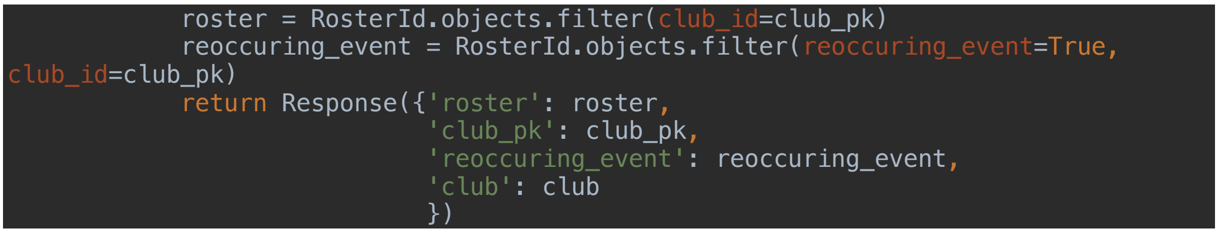
```
class PitchRental(APIView):
    renderer_classes = [TemplateHTMLRenderer]
    template_name = 'pitch_rental.html'

    def get(self, request):
        club_pk = request.session.get('pk')
        club = ClubInfo.objects.filter(pk=club_pk)
        inital_data = {
            'club_id': club_pk
        }
        form = RentalForm(initial=inital_data)
        form.fields['pitch_id'].queryset = Pitch.objects.filter(club_id=club_pk)
        return Response({'form': form,
                         'club_pk': club_pk,
                         'club': club
                         })

    def post(self, request):
        club_pk = request.session.get('pk')
        club = ClubInfo.objects.filter(pk=club_pk)
        form = RentalForm(data=request.data)
        form.fields['pitch_id'].queryset = Pitch.objects.filter(club_id=club_pk)
        if form.is_valid():
            form.save()
            return render(request, 'booking_complete.html', {'club': club})
        else:
            return Response({'form': form,
                             'club': club
                             })
```

The above code class is used to retrieve and display the form to the end-user to complete pitch bookings. The get method uses the session key called 'pk' which is initially created when a club is selected from our clubs page. This 'pk' is used to populate the form to ensure valid data is being committed to our database. We also filtered out the available pitches associated to the club 'pk' to create a drop down list of pitches related to that club. The variables are passed into a dictionary so that they can be utilized in our html files. The post method validates the form and posts it to our database.

```
# Model to store pitch information to be used for rental and roster purposes
class Pitch(models.Model):
    club_id = models.ForeignKey(ClubInfo, on_delete=models.CASCADE,
related_name="pitches")
    pitch_name = models.CharField(max_length=30)
    PITCH_SIZES = (
        ('S', 'Small'),
        ('M', 'Medium'),
        ('L', 'Large'),
    )
    PITCH_TYPE = (
        ('1', 'Outdoor'),
        ('2', 'Indoor'),
    )
    pitch_size = models.CharField(max_length=1, choices=PITCH_SIZES)
    pitch_type = models.CharField(max_length=1, choices=PITCH_TYPE)
    open_time = models.TimeField(default='09:00')
    close_time = models.TimeField(default='22:00')
    RENT_TYPE = (
        ('0', 'Not Available To Rent'),
        ('1', 'Available To Rent'),
    )
    rental = models.CharField(max_length=1, choices=RENT_TYPE)
    rental_price = models.DecimalField(default=0.00, max_digits=6,
decimal_places=2)
    max_people = models.IntegerField(null=True)

    def __str__(self):
        return self.pitch_name
```

The above source code is an example of a model for clubs to store pitch information. It contains a foreign key to the club id. Accepts multiple choice inputs regarding pitch information e.g. size, type and rental availability.

```
# Class to handle roster information
class ClubRoster(APIView):
    renderer_classes = [TemplateHTMLRenderer]
    template_name = 'roster.html'

    # Get method to retrieve current roster information and form
    def get(self, request):
        if request.user.is_authenticated:
            club_pk = request.session.get('pk')
            club = ClubInfo.objects.filter(pk=club_pk)
            club_info = ClubInfo.objects.filter(user=request.user).first()
            reoccuring_event = RosterId.objects.filter(reoccuring_event=True,
club_id=club_pk)
            inital_data = {
                'club_id': club_info,
            }
            form = RosterForm(initial=inital_data)
            form.fields['pitch_id'].queryset =
Pitch.objects.filter(club_id=club_pk)
            form.fields['team_id'].queryset = Team.objects.filter(club_id=club_pk)
            roster = RosterId.objects.filter(club_id=club_pk)
            return Response({'form': form,
                             'roster': roster,
                             'club_pk': club_pk,
                             'reoccuring_event': reoccuring_event,
                             'club': club
                             })
        else:
            club_pk = request.session.get('pk')
            club = ClubInfo.objects.filter(pk=club_pk)
```



```
            roster = RosterId.objects.filter(club_id=club_pk)
            reoccuring_event = RosterId.objects.filter(reoccuring_event=True,
club_id=club_pk)
            return Response({'roster': roster,
                             'club_pk': club_pk,
                             'reoccuring_event': reoccuring_event,
                             'club': club
                             })
```

The above code sample is used to retrieve roster information. Information is filtered by the club pk session key. The if else statement is used to hide aspects of data from registered and non-registered users. In this example we hide the form from non-registered users as they do not have the ability to add roster data. Additional validation is completed in the html file for extra security.

# DCU School of Computing
# Assignment Submission

| | |
|---|---|
| **Student Name(s):** | **Conor Ward, Gavin Boyle** |
| **Student Number(s):** | **14316061, 14312781** |
| **Programme:** | **Bachelor of Science (Hons) in Enterprise Computing** |
| **Project Title:** | **Enterprise Computing Team Project** |
| **Module code:** | **Ca472** |
| **Lecturer:** | **Renaat Verbruggen** |

## Declaration

**I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying is a grave and serious offence in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion, or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references.**

**I have not copied or paraphrased an extract of any length from any source without identifying the source and using quotation marks as appropriate. Any images, audio recordings, video or other materials have likewise been originated and produced by me or are fully acknowledged and identified.**

**This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. I have read and understood the referencing guidelines found at http://www.library.dcu.ie/citing&refguide08.pdf and/or recommended in the assignment guidelines.**

**I understand that I may be required to discuss with the module lecturer/s the contents of this submission.**

**I/me/my incorporates we/us/our in the case of group work, which is signed by all of us.**

**Signed: Conor Ward, Gavin Boyle**