2021

Software Projects: Project Proposal

AN ONLINE WEB APPLICATION FOR FINDING FOOTBALL TEAM OPPONENTS IN YOUR AREA

HAMSA IBRAHIM (EMAIL: HIBRA003 @CAMPUS.GOLDSMITHS.AC.UK)
THOMAS MCLOON (EMAIL: TMCAL001@CAMPUS.GOLDSMITHS.AC.UK)
MOHAMED ISMAIL (EMAIL: MISMA005@CAMPUS.GOLDSMITHS.AC.UK)
MUSTAFA AHMED (EMAIL: MAHME062@CAMPUS.GOLDSMITHS.AC.UK)

Contents

Introduction

Introduction

The main concept of this project is to provide a web application-based system for Sunday league matches. The customers would mainly interact with this system through a tablet device, phone , computer. The web app interface is very simple for customers to use, they would be able to book a football match game without any hassle. Also, this web application is very detailed, it will contain a league table with matchday stats, so users will be able to see their team progress after finishing a match.

Aims and Objectives

Having a set of deliverable components is important in a project because a deliverable is anything that is produced and provided because of a process. For example, when the goals are met then the deliverables are produced, when the project is fully completed the key deliverable will have been created. A key deliverable is something that has been produced or is provided as a result of a process. When the aims and goals of the project are met then the deliverable is produced. At the start of any project that is being created there must be an end goal of what needs to be achieved, and there must be a clear plan on how to achieve that goal. Also, a project manager can lay out the timeline of deliverables that need to be met at certain intervals. There’s an important difference between deliverables and milestones, which means for a project milestone you don’t need something that’s delivered to customers and internal stakeholders. That means they are checkpoints for your project. To be able to define key project deliverables in your project it's best to start by identifying your objectives so that you can define the certain tasks that need completing. External deliverables are those you would convey because of a project. It would be something that would fulfil the client's need. So that means the actual business or motive for what the project is being executed.

There 2 types of Deliverables, the Tangible deliverable, and the intangible deliverable. Deliverables can either be intangible or tangible, an example of this is that a tangible deliverable will be a construction of a new headquarters to place workers that couldn’t fit into the old headquarters, or it could be a brand-new factory that needs to be built so that it can boost production levels. Another example of an intangible deliverable would be a football league web app for Sunday league players so that they play against players that are at their level, also there would be basic instructions at the beginning on how to use the app and book matches.

Essentially a project timeline is a list of deliverables. A detailed timeline provides an overview of a project from start to finish. It determines milestones, dependencies, delivery dates and tasks. There are many dependencies in projects an example of this is that some projects that are turn based games (games that allow players to pause in game) that's means they will have to wait until a certain action is completed to be able to move onto the next task in the project. Having lists of tasks for each milestone, you need to give out precise orders that must be followed, also 1 task is dependent on the other task, that means some tasks can only be ready if the other task were to be completed. This is extremely common for IT firms, where the developers cannot start working on the next part of the project without their colleagues finishing off theirs. Estimating how long each task will take help finish completing the project timeline, also project managers can suggest how much time will be needed to finish off a task. It can be measured in hours or in points. However, project managers have to be really accurate when estimating tasks, because it has a huge influence on the whole project plan, also its much better when you include your project team members so that an absolute estimation can be made in place.

The important thing here is that you're also going to be other members know how you're doing on the project. Those milestones can help you make it clear to show just how far you've come and how much further you must go. By looking at the milestones that you've achieved your team can see the progress and so can a client. They can see that you're moving forward and that you're putting a lot of time and effort into their project rather than just waiting until the last minute. These milestones can be deliverables, or they can just be signposts that you use along the way to mark your progress.

The Stakeholders

How we carried out data gathering about your user groups

The most efficient way is to put users in the same groups if they have similar aspects, this means that we will be able to see what different groups of people that come under the same criteria have in mind when we gather data. Doing this will help us see what individuals like the most, furthermore allowing us to improve our software in the process by receiving feedback.

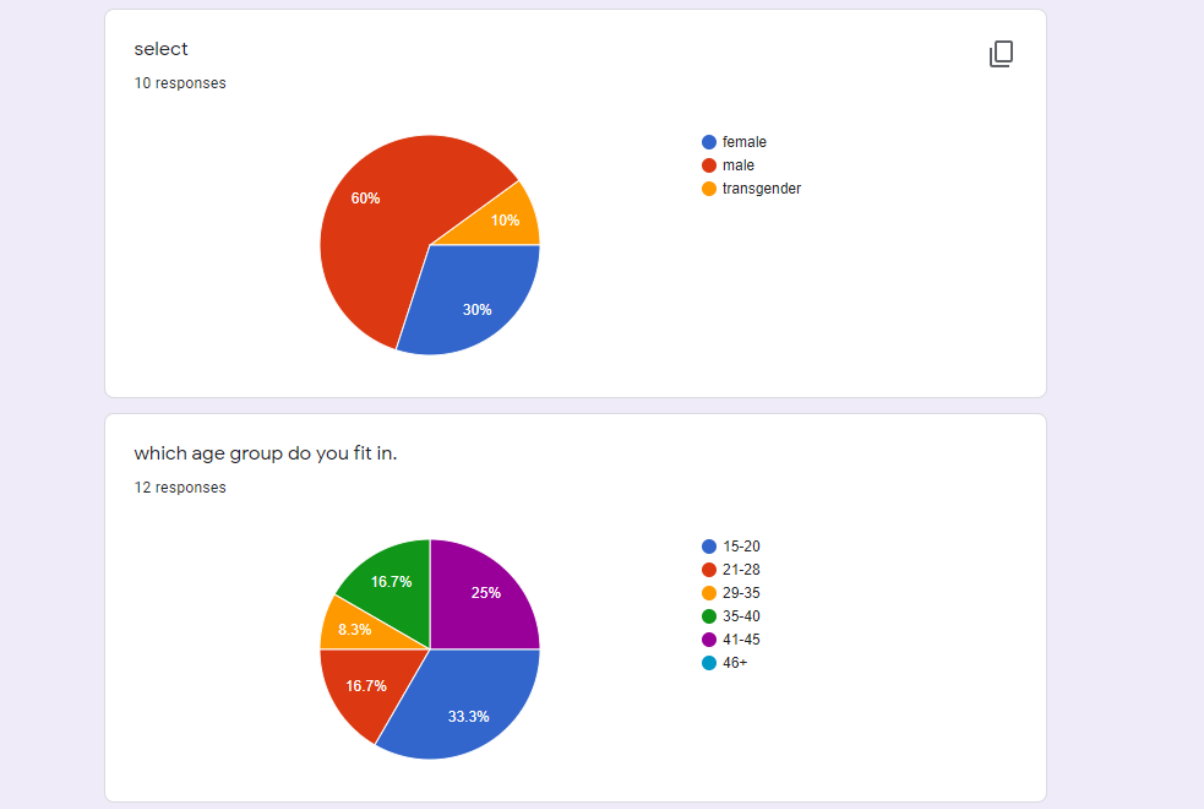
Certain user groups will be able to help benefit in certain parts of the project, therefore allowing us to take information from specific user groups in order to implement them into the project. Benefits of this is that we take notice of the user's opinions and will take act on the information allowing the project to be adaptable to all the user's needs.

Requirements for the proposed design

Functionality

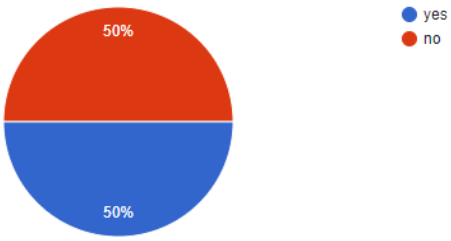
In software engineering functionality is where a software must perform, this means that the software is meant to work on the intended system that it has been designed for. This is important because the app not being functional will affect the usability of the software thus meaning that the app will not have as many users. It is important that the app will be able will be able to run on old system allowing people to run it on phones with optimal performance will mean that the app will have to be coded so that if its run on a phone's website it will be able to change it into so that it will be able to fit the phones screen.

User characteristics



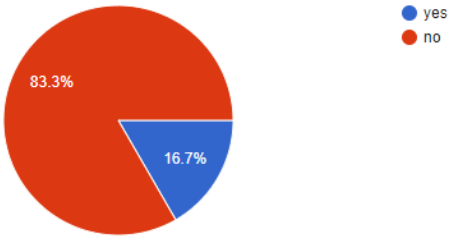
do you play football?

10 responses



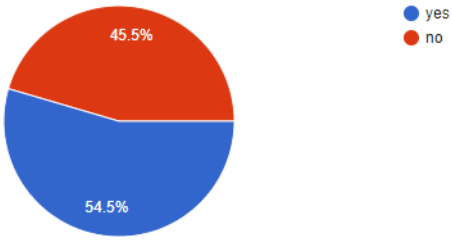
do you gamble?

12 responses



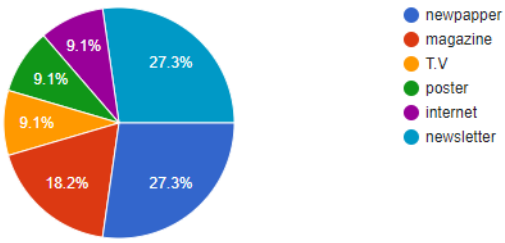
Have you ever played in a tournament?

11 responses



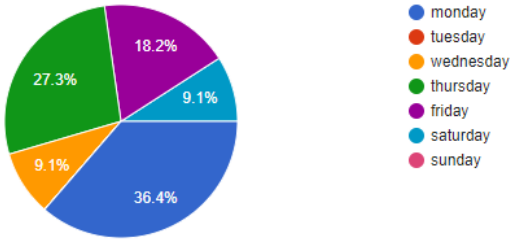
Where did you see this advertisement?

11 responses



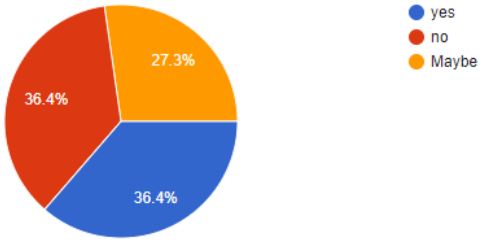
what days?

11 responses



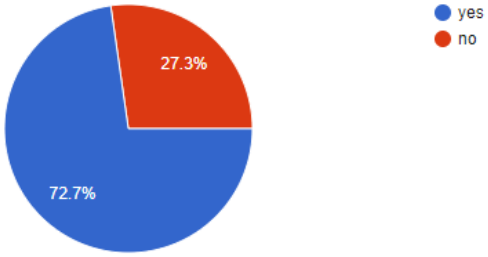
would you say football is your favourite sport?

11 responses



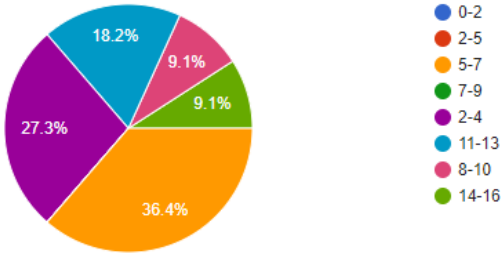
would you participate?

11 responses



how many hours a week do you play football?

11 responses



Accessibility issues

within the website I have to ensure accessibility in many different ways, for example, we will have ensure that the text is visible and by this, we have made sure that we have a suitable background for it by having a dark background and light text. we have ensured that it is accessible by asking individuals if the text is readable on the website. furthermore, we have made sure that the size of the text is also readable by choosing an appropriate size not only for it to be accessible for people to read so that it also looks nice on the website.

another accessibility that we must ensure that people will be able to expand the text of the contact box so that they will be able to write as much as they wish depending on what they want. this ensures to keep the customers satisfied with what they need. by ensuring that the accessibility is functional we will get people to try it out and get feedback on whether it was good and resourceful when an individual wants to write a bit more so that it will be fill for their needs, furthermore, having this is an advantage because it adds to the design of the website as it makes it have a unique and simplistic design. this, therefore, means that individuals will be able to send a detailed message within this message box allowing them to give them their need into what they wish.

Usability goals

The main reason as to why usability is important not for us but for everyone is because that it is important for people to navigate easily around the website without having trouble. For example, they will be able to book matches without having an issues to find the booking section this is important because if they struggle to do this they may find another website which then will affect us. This allows us to help bring more customers as better looking websites will be able to bring the competitive edge over our competitors.

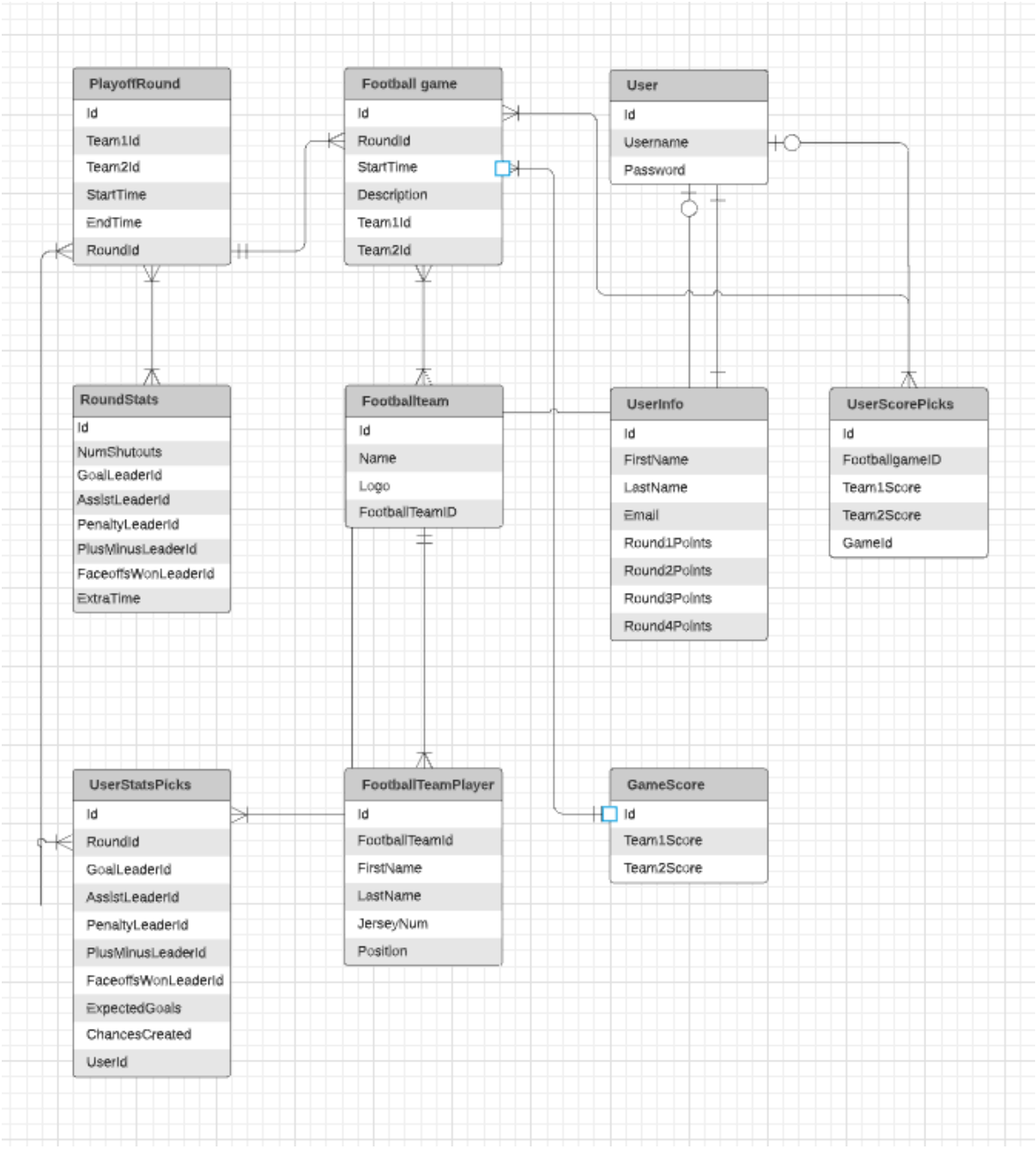
User experience goals

It is important that users will be able to use the app without having any issues, by this I mean that we would want the users to be using the app for what we made it for and for them to not have any issues is very important. This means that, we will as for reviews on the app to see whether if they users have any issues with it, by this they will be able to submit their opinions about the app thus helping us improve it for everyone's use. This is important because if we don't enhance the users experience to the users' standards the issue will keep coming back once more users complain.

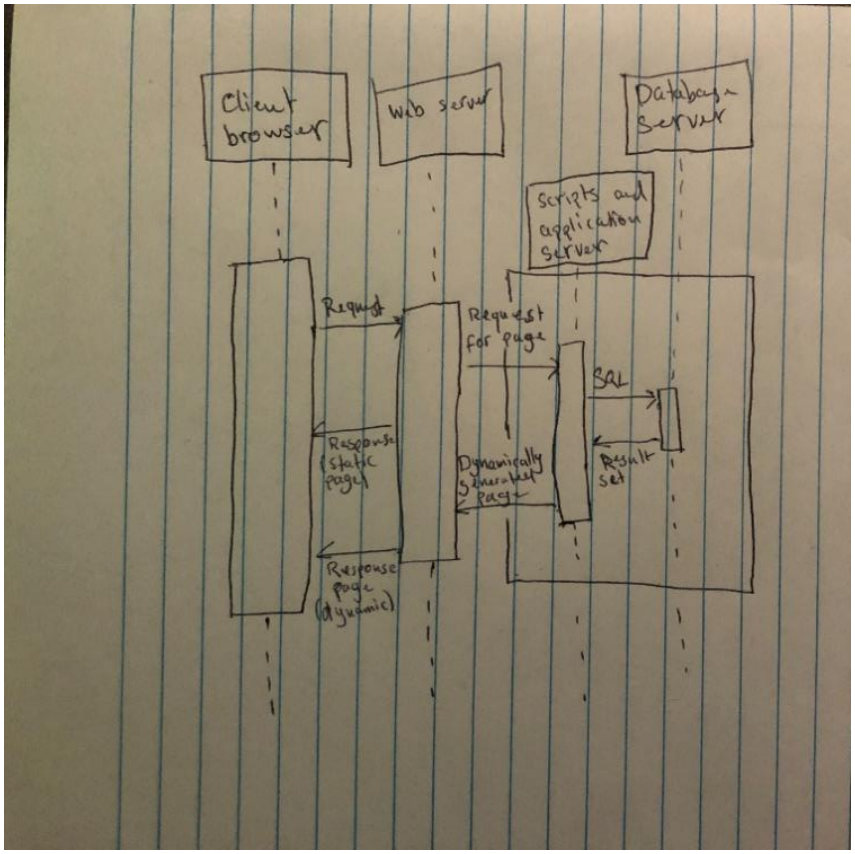
Project management

Formal specification of the desired system

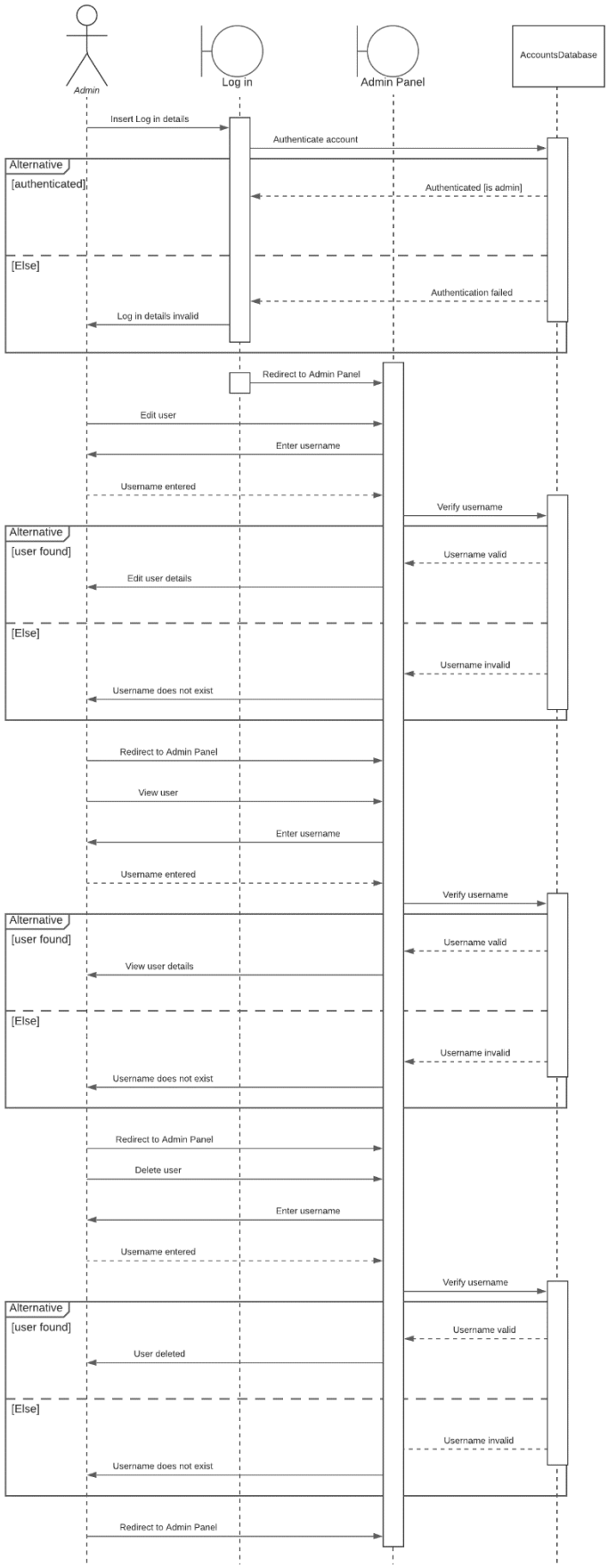
UML Diagram



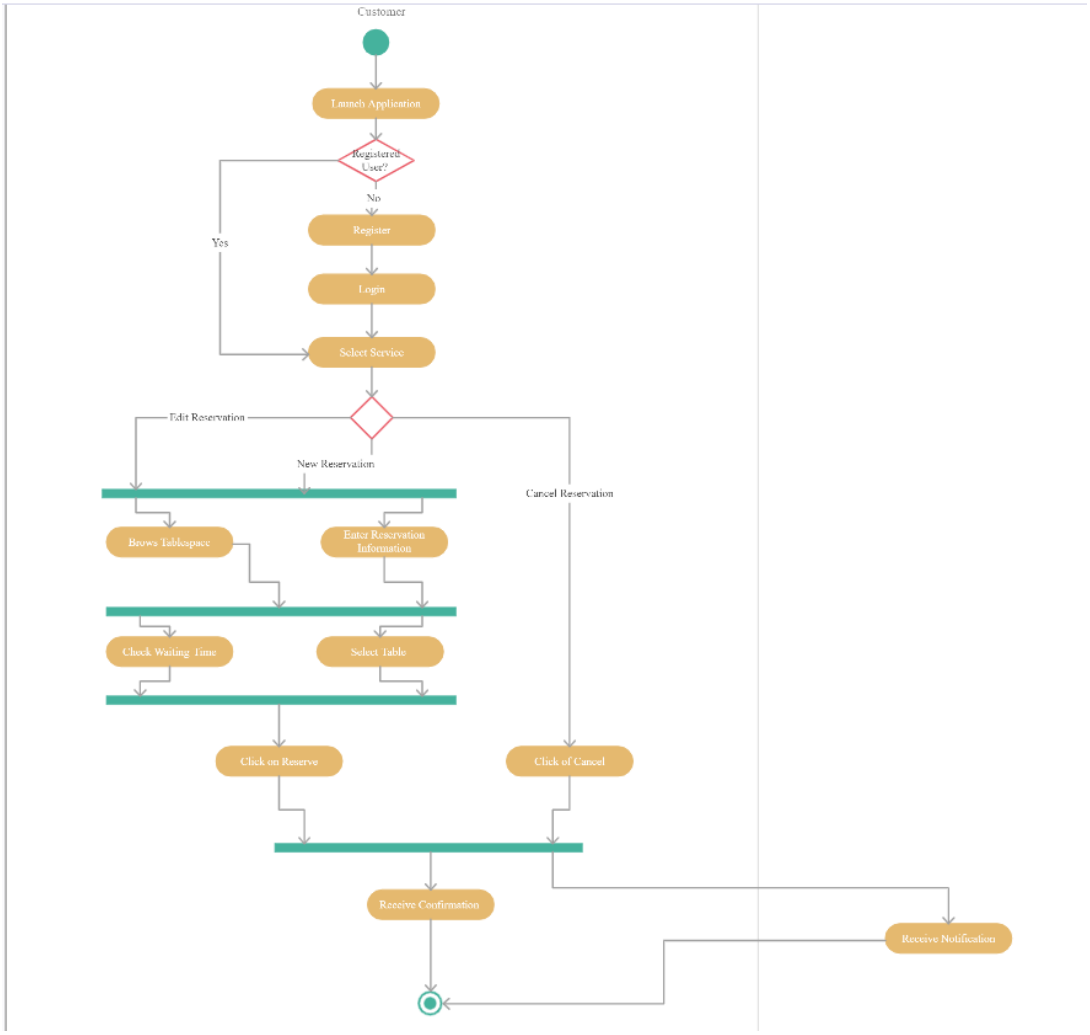
Sequence diagram for user:



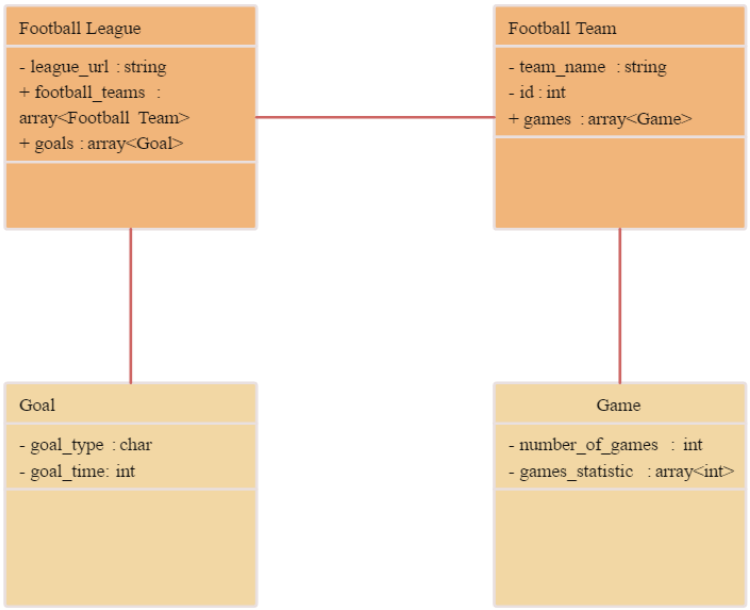
Sequence diagram for admin:



Low quality sequence diagram:



Low quality uml:



Functional specification

the functional specification of the application will be divided into different sections. This will be arranged mainly by user, admin. The functions specification is for us as the project planners to know what we need to achieve in the system allowing us to focus on the main part.

Technical Architecture

Our group have decided that we will be going to us HTML, CSS and JavaScript in order to make the web application. This will be able to be adaptable to users who use tablets and mobile phones.

An architecture assists a business in analysing and identifying areas where modifications may result in cost savings. For example, the architecture may demonstrate that numerous data base systems may be altered such that just one product is utilised, lowering software and maintenance expenses. It may discover that decreasing complexity and standardising on a small number of desktop computers can cut support expenses. Analysis of the basic architecture may reveal that alternative standards are useful (on the other hand a good analysis can show when standards would not be beneficial and should be delayed or avoided). A crucial guideline may be to search for methods to optimise internal and contract workers rather than technology per se Since people cost more than most technologies.

System Requirements Specification

1. *Purpose*

The use of this app will be that users will be able to book matches and competitions, theses competitions will have a prize for the winners, the audience will be for people who would like to compete and for those who play football only. In the future when this app generates more revenue, we will then plan to expand it to different sports, therefore allowing more of a variety for the users.

2. *Scope*

This project has involved researched and by doing this we have looked a several different competitors, ones that host football tournaments with money involved and ones that do it for charity and ones that only organize matches and tournaments for no winnings at all. This project will try to adapt to the most part of it, therefore meaning that later in the project we will try to adapt it for charity games.

3. *Functional requirements*

User side:

- Be able to change languages
- Add themselves to teams
- Pick a date and time

- Be able to join different games that are open
- Add extra players
- Click confirm
- See the prices
- Leave a review
- Information about the game
- Read the rules

Admin side:

- Edit teams
- View reviews
- Check different matches
- View game id

4. *Non-functional requirements*

Performance – site should not take no longer than 3 seconds to navigate on by using CSS will help load the information faster than using figures.

Reliability – the web application should be able to be reliable and have fast and easy access.

Security – admin will be able to inspect the security of the website which will allow for it to be a secure system.

Maintainability – the site should be maintained by the admin.

Capacity – this will grow as more users use the app.

Data integrity – it will be ensured that the data stored on the SQL server is structured and consistent.

Ethic Audit

Within our application we would be collecting low value data such as names, Date of Birth and other football related statistics. Although this data is not very valuable to hackers and online criminals, we are still very determined to keep it safe. To do so we will be using MySQL since it very easy to use and maintain. It also has very tight and trustworthy security.
During our project we will not directly be working with minors or people with disabilities, but this does not mean that our web application is not available to them.

Formative Testing and Evaluation (Mustafa)
5) Some evidence of requirements elicitation involving some/all of your project stakeholders.

Initial approach to stakeholders & user needs

In order for us to create an application that perfectly fits our wide range of user needs, we were required to do copious amounts of research. During the early planning stages of our application, we contacted 4 different Saturday league football teams about potentially becoming stakeholders for our web app. Thankfully they were all willing to be a part of our project.

Demonstrating our app

After our initial encounter we called together our stakeholders to demonstrate our working prototype. We gave our stakeholder the opportunity to test the app for themselves and later on asked for what we can improve on or add to ensure our application fit the requirements they needed.

A stakeholder Zakaria Ali, proceeded to point out a specific feature they would like us to include within our app. He pointed out that our application does not include any information about teams other than their name and location. Users should have the choice to select opposing teams and be taken to a page which includes specific information about each team. Information such as wins and losses, age and starting players. This would allow teams to have the choice to choose a team on more than the distance they are away from one another. This information would also allow team to pic opponents that are at a similar skill set. We have taken this into consideration and have decided that this feature would be very useful to our users therefore, we have included this into our web app.

After we had presented our first prototype, some shareholders had a concern with our website not being able to be used as an app that users could download from their phone. They pointed out that this would eliminate the hassle of them having to search up our website every time they need to use it. Our group took this point into consideration and attempted to create a web app that people could have on their phones and also search up on computers. We eventually decided that with the time period given for this assignment and the little knowledge we had about mobile apps, this extension to our projects would be unachievable. We continued to develop our initial idea.

During developments, we also encountered another stakeholder by the name Marcus Clarke. This stakeholder was very intrigued by this idea and also liked the format of our web application. He then proceeded to point out that we have neglected to include a feature that allows users to find football pitches in equal proximity of both teams. Our group tackled this problem by including a list of available pitches and links to their websites to users once they input their location and pick an opposing team. Once we included this feature into our application, we

introduced it to our stakeholders. They were very pleased with the outcome and prototype app.

Summary of stakeholder interaction

After our interaction with our stakeholders, we proceeded with the development of our app keeping in mind the suggestions given to us by our stakeholders. As previously stated, we were unable to include all features presented to us due to the time period given to us. Our development went forward smoothly, and our app has proved to complete its designed task. Our application runs on pc platforms but is also accessible through web browsers on the mobile phone.

We have gone through our code and thoroughly tested our app to ensure it is capable to complete the task that it has been designed to do. Through this, we have created a fully functional application that our stakeholders are very impressed with.

We have performed small tests using local football teams to filter and fix all bugs. Initially, a problem we faced with our app was that users were unable to input specific locations and the app randomly assigned teams. The was because of technicalities with our code. Thankfully this was fixed
Another problem we faced was that our app was unable to show users a summary of their activities on the web app. Users would select opponents and would be kept on the same page with no information on what has happened. To fix this problem we included a page that users are redirected to once they have selected an opponent. This page shows information such as the opponent the user has selected, and all other key information needed. This page is named the summary page.

Challenges of working within your chosen domain

Due to recent problems occurring in the world such as covid, we have struggled with many things in the development of our web application. Due to these problems, things like finding stakeholders have become very difficult especially in our chosen domain. Our application depends on the ability of people being able to come together in large quantities and physically interact by playing football.

Covid rules have made this very difficult. Rules such and no social gatherings and lockdowns have made it nearly impossible for people to interact on our application. From March 2020 until July 2020 the government had made it illegal to socially gather with people outside of our family or support bubble. Fines and even jail time were the consequences for breaking these rules. Situations such as these have made applications such as ours dormant.

The Model Design

Software Development Models & Methodologies

It is important to assess tried and tested prototyping techniques and decide which is most suitable for our project. This section of our proposal will discuss three models we considered in the prototyping phase of our project. We will explain how we will implement the techniques into our prototyping and how they will aid in the development of our project.

Agile Methodology

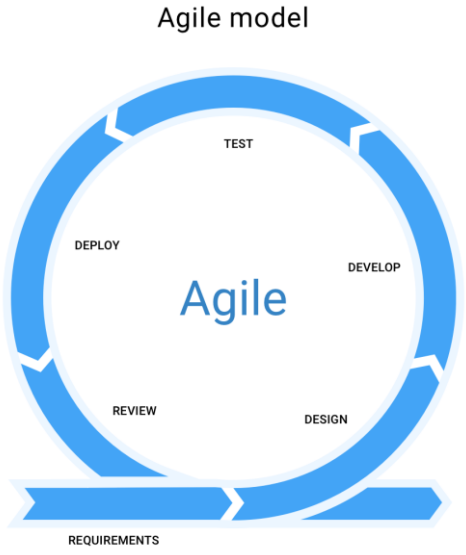
Agile Software Development focuses on four fundamental, core values that underpin solid software development.

1. Individuals and interactions - Rather than solely emphasizing systems and tools, the focus should be on the people within the team and the interactions they have while working together on the project.

2. Working software - focus on producing a working product, or even a simple prototype, that conveys the design goals, or the components used throughout the application

3. Customer collaboration - it is crucial that the project is constantly open, willing, and able to respond to customer feedback and behaviour.

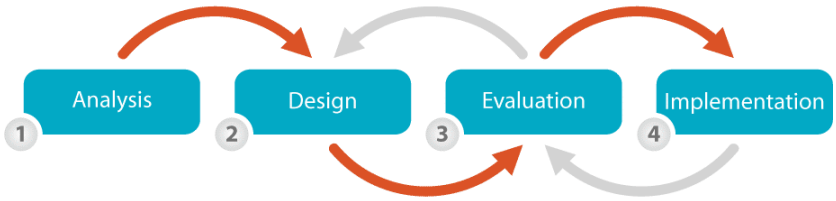
4. Responding to change - the ability of the project to adapt and respond to the ever-changing needs of everyone and everything involved.



(Peterheria, 2018)

User-Centred Design

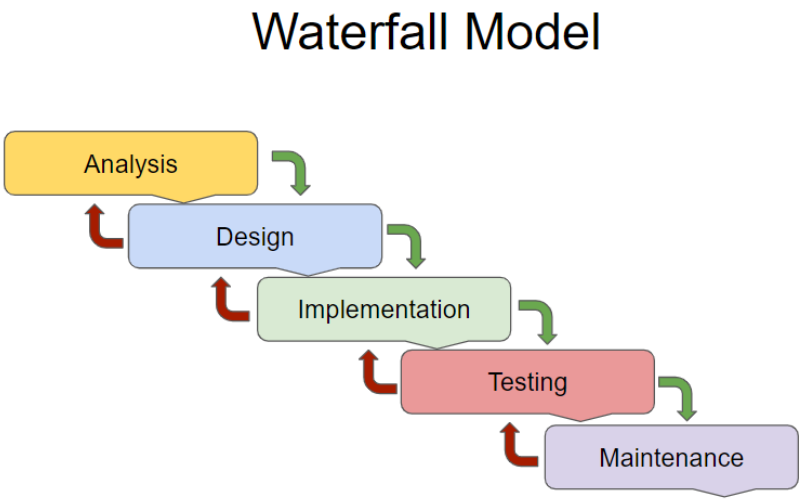
User-centred design is an iterative design process framework that involves validation from the user repeated throughout development. This frequent interaction aids in understanding the user, what they want and don't want from your design, and how they're ultimately going to interact with each part of your product.



(User Centered Design, n.d.)

Waterfall Model

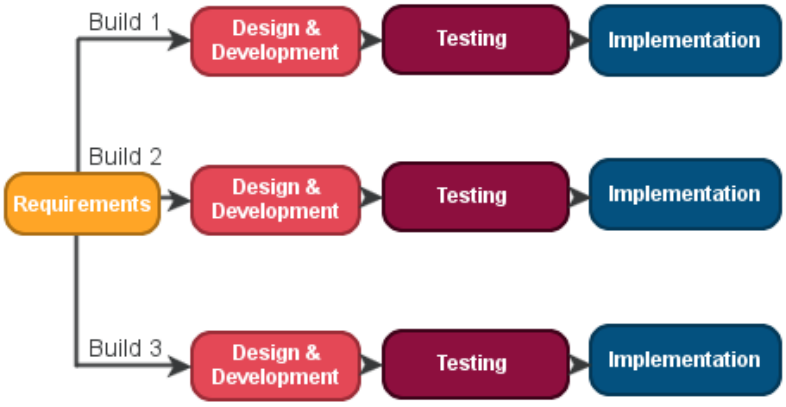
The Waterfall model is a sequential process model which does not overlap which means that until one phase is not completed then the next phase cannot start. As it is simple and easy to understand the is a strong model to use.



(Development Life Cycles, n.d.)

Iterative Model

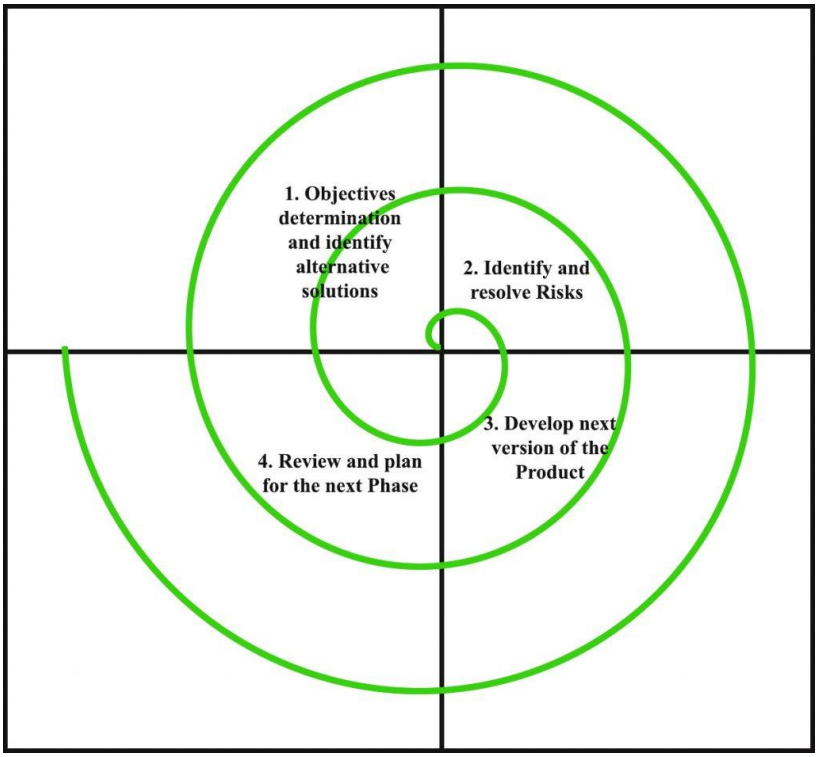
In the iterative model, each iteration (build) evolves a small, varied version of the product, which is repeated until the final version is developed. The Iterative process model focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. New functionality is added continuously until it is completed.



(What is Iterative Model?, 2019)

Spiral Process Model

The Spiral Process model is a combination of the Iterative Model and a sequential model like the Waterfall Model. This model is favoured by software engineers for large scale complicated and expensive projects due to its ability to manage unknown risks after the project has commenced; creating a prototype makes this feasible.



(Pal, 2021)

Analysis of the Spiral Process Model

The Spiral Process Model could be beneficial to our project as it offers flexibility, risk assessment and customer satisfaction. Changes made to the requirements after development has started can be easily adopted making the Model flexible to our needs. As the Spiral Model involves risk analysis, which can improve security, reduce the risk associated with the projects (such as users' information being leaked) and breakages. The spiral model facilitates customer feedback. The intended user will be able to see and evaluate our product in every phase, therefore, allowing them to suggest changes before the product is fully built, saving our group time and money.

However, there are drawbacks to this model. The spiral model is expensive furthermore making it unsuitable for small projects. Also, the spiral model is complex and for it to operate efficiently, protocols must be followed closely. It can be hard to manage time meaning there's always a risk of our group falling behind schedule.

Our Development Model & Methodology

When choosing our development model/methodology, we found elements from the Iterative Model and Agile Methodology worked in synergy with one another which we believed would work best for our group. Therefore, we have created a hybrid development model that'll guide us through our development efficiently.

The model implements elements from both the Iterative Model and Agile Methodology, the reasoning behind this decision is that the two complement one another. Agile software development allows us to release our software in iterations. Iterative releases will improve our groups' efficiency as it allows us to find and fix defects and evaluate our expectations for the project. We understand the importance user feedback will be to the successful development of our software therefore, the importance of Agile Methodology can't be stressed enough as it facilitates sooner use software by the user, with frequent incremental improvements through their feedback.

Our model will include the Iterative Model as it is ideal for our project, the final product will be more tailored to the user because of this model. Constant interaction with the target market will lead to valuable feedback which will help us develop an application with desired features and a better user experience. Another reason behind using this model is to find functional or design related flaws as early as possible, which is vital when considering our budget and time scale, we cannot afford to waste time or resources on an unfeasible design. Concerning this, our model will spend more time on design rather than documentation. Our model will accommodate changes in requirements, which will allow for contingencies throughout the project.

Prototyping Techniques

Software Prototype Techniques

- Throwaway Prototyping - The prototype is expected to be short term, such as one sprint in the Agile development framework. It may go through several cycles of feedback, modification, and evaluation during that time.

- Incremental Prototyping – The prototype passes through requirements, design, implementation, and testing phases. update of the software adds function to the previous version which continues till the complete system is achieved.

- Horizontal Prototyping – The prototype is usually developed during the early stages of analysis. It typically contains all the system functions on menus, but includes only dummy screens, reports, and database queries (if applicable) for core functions.

- Vertical Prototyping – This prototype is the back end of a product used to improve database design, test key components at early stages or showcase a working model, though unfinished, to assess the functionality.

Understanding what a Prototype is and why it essential to our project

A Prototype is a model to demonstrate a concept, it is intended to be developed from a basic visualisation of an idea to a final product to be deployed. The Prototype can vary in complexity and completeness and does not always hold the exact logic used in the actual software application; however, the importance of prototyping cannot be stressed enough as it is arguably 'the first step to success' (Mishra, 2019). Prototyping allows the users to evaluate a concept and try it out before implementation and final deployment.

During our software development process, prototyping is the ideal way to test, evaluate, and validate our idea. It lets us confirm that we are building the desired product before we code it, in theory, prototyping reduces the risk of a failing project.
We can make informed decisions based on an early and quick prototype (low-fidelity prototype) before spending time and resources on the development of our final product. User feedback can also be given on more developed prototypes which is essential to the success of the project.

Stages of prototyping & how we will build a Prototype

1 – Assessing Requirements of the Software

Assessing the requirements of the software is the first stage in developing a prototyping model. During this phase, the targeted demographic of the product are asked what they would like to see implemented in the software and what needs they want the software to cater to.

In this project we will conduct our requirement assessment through the use of surveys, we will use Google Surveys as they are quick, and Google makes the interface easy to use and user friendly which is important as we need quality market research for our project to be successful.

2 – Low Fidelity Design + Feedback

The second stage involves the creation of a low fidelity design or a quick design. Throughout this stage, the basic design for the system is formed. It provides the user with a quick overview of the system. This quick low-fidelity design assists in the development of the high-fidelity design/prototype.

The reasoning behind using low-fidelity designs is that they are quick and inexpensive which is important to our group as we must stick to a timescale, and we have limited resources. Possible to make instant changes and test new iterations. They are also disposable which is convenient if you make a mistake or would like to alter the design. Also, regardless of ability and experience, stakeholders and users can get an understanding of the concept.
Most importantly a low fidelity design enables us as a group to get an overall view of the project using minimal time and effort, instead of wasting time on minor details.

We will produce a low fidelity design by hand, and it will show the basic design of the website UI, the UML Diagram which will show an idea on how the database will work and the backend design such as communication between the website and database. This is an essential step for us in the development of our project as it ensures we work towards what the user wants. Once the low-fidelity design is created, we can get expanded feedback from the target demographic and get an insight into the reaction of the user to our vision.

3 - Medium Fidelity Design

A Medium-Fidelity Design focuses on the overall design and concepts for the proposed software without the pressure of making every page linked, clickable, and interactive, in other words, we can spend less time worrying less about the bugs of the project and focus on the ideation of the project.

We will also have a more developed UML diagram and a further developed database itself being used within the prototype. This will be the first time we thoroughly assess what languages and systems will be used to build the project, for example, the choice to use React over Angular or MySQL over Postgres. The prototype will be interactive which can improve the collaboration, as a group, this stage will help us to understand one another's ideas and work them into the project.

4 – High Fidelity Design (Functional Prototype)

During this stage, an actual prototype is developed intended to support the knowledge gained from the low-fidelity design. It is a simple low-level working model of the desired system. We will show this to potential stakeholders interested in the project as well as users to get continued feedback to implement in later prototypes.

It is important to find a balance of building something that shares an understanding of what the final product will be to the user and stakeholders without wasting time and resources by overcomplicating this original prototype. We will create a functional database that works alongside a website to demonstrate how the user will create an account and join a club among many other features. The main aim will be to have a tool to demonstrate our project to potential stakeholders and users rather than working on the final design of the software, therefore the UI will remain basic and will be addressed in a later stage of the prototyping. The languages and systems used to build the project will be finalised and the prototype will be developed using the finalised languages and systems.

5 - User Evaluation

The proposed software is demonstrated and tested by intended users, it is vital we investigate the prototypes strengths and weaknesses as these will be focussed areas of development. Detailed feedback from the intended user and potential stakeholders is gathered to be implemented. This process tends to run in a circle until we get overall satisfaction from the user where a final product can be deployed.

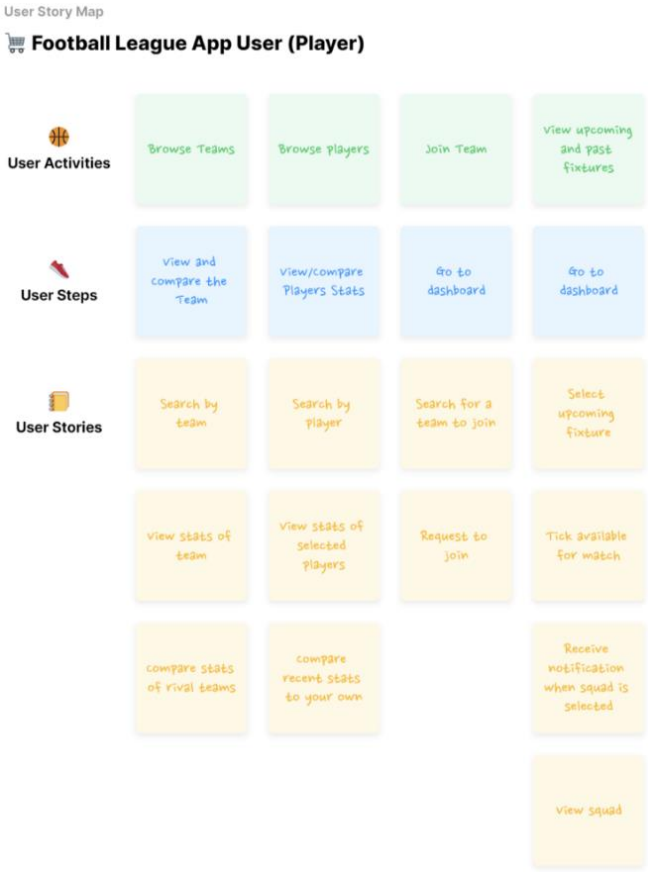
To carry out this stage we may consider getting a focus group of intended users and giving them a spreadsheet to give feedback on any issues they encounter (bugs) or any features they would like to see implemented. We can then use this spreadsheet as a checklist to work through while developing the prototype.

6 – Implementation of User Feedback on High Fidelity Design

If the user is dissatisfied with the current state of the product, then we will want to improve the model so that it responds to user feedback and suggestions.

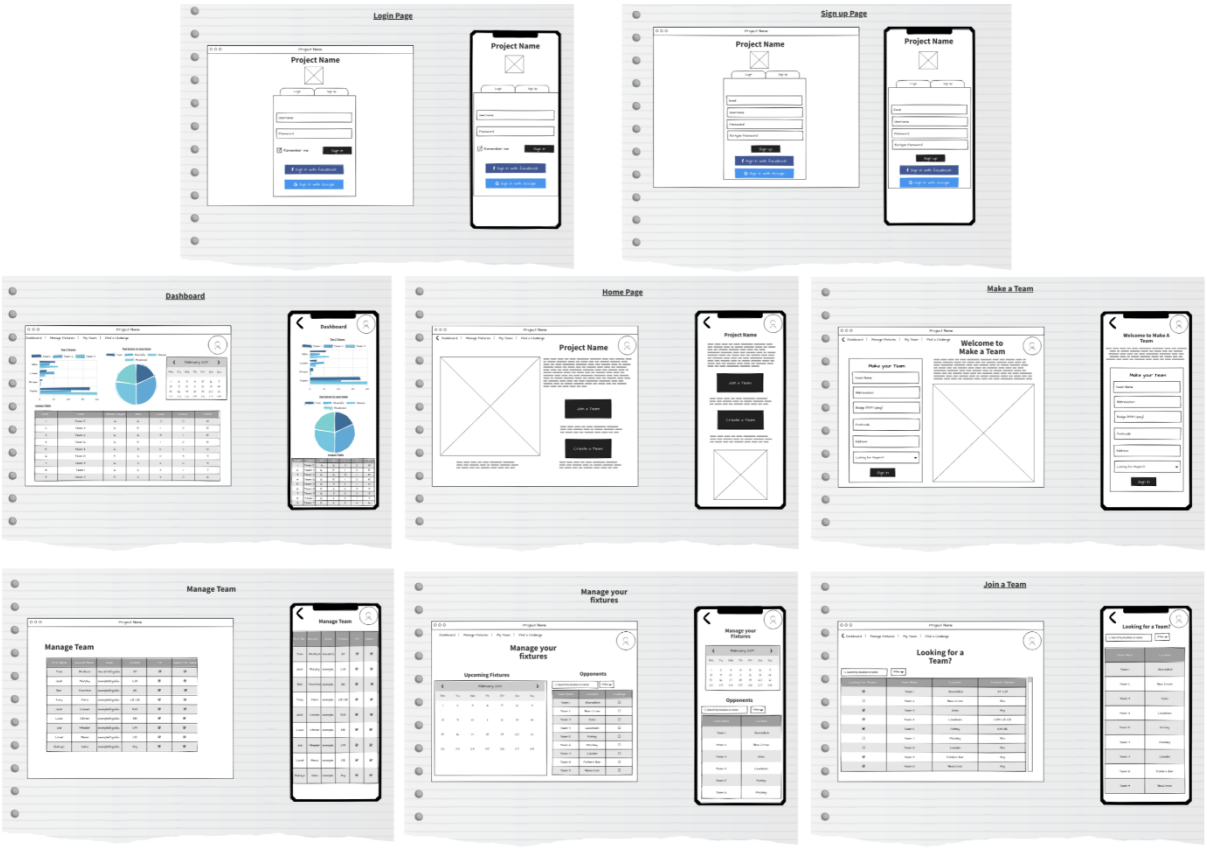
This is the final stage of our prototype; we will use the user feedback from the high-fidelity design and implement their feedback into our project. Once the user is happy with the product we can move on to the deployment of our software, the prototype stage of our project would now be complete.

User Story Map



Our Prototype

Low Fidelity Design



This is an overall screenshot of our low fidelity design that was created using (https://wireframepro.mockflow.com/, n.d.). For a more detailed view please refer to the Appendix (C-J).

Feedback on Low Fidelity Design

“Are we going in the right direction?” was the main question to be answered in this feedback session especially as we were at the early stages of the design process. To find an answer to this question, we created a low fidelity prototype.

Using a free software tool online we were able to create a basic sketch up a wireframe of what we thought our product would look like. The purpose of this prototype was to get an understanding of how we would lay out our product and whether it was a feasible design proposal. We understood that it wasn’t aesthetically pleasing, however did showcase our vision to our intended users. Our goals and considerations when testing:

- **Test the right people**
  We tested seven people that we believe would fall into our target audience, the demographic of those tested was 19–24-year-olds that play football anywhere between frequently (weekly) to occasionally (handful of times a year).

- **Be neutral when presenting your ideas**
  We needed not to sway the opinion of the tester in any way, we made sure to present ideas without a biased opinion. This ensured our feedback was authentic and reliable, making it far more valuable to our future iterations.

- **Remain Open-Minded**
  On our testing feedback sheets we implanted an “I Like, I Wish, What If” method to allow participants to share their feedback in a critical but positive manner. Also, we made sure not to respond to negative feedback defensively but instead to ask useful questions to understand what exactly is wrong with our design, and how we can improve it.

- **Communicate limitations to the user**
  As this was only a low fidelity design, we made sure to inform the tester that the focus was design feasibility, “could you see it work or is there anything you would change?”, “anything you particularly dislike or like?”. As previously stated, we know the design needed to be more aesthetic however we told the tester this would come at a later stage.

- **Ensure to Iterate**
  Going back to our development model and methodology, we understood that we would have to use the feedback to produce an improved iteration of the original design. We decided to sort negative feedback and suggested improvements into an order of importance (logic and architecture before aesthetics).

Tester	Keep Doing +	Improve -	Flight
Jake	The calendar is a nice feature	Some pages have a lot going on	A to do list would be nice
John	I like the tables, they’re easy to read	The links at the top aren’t big enough or helpful	Seeing what players are ready to play should be included
Evan	The dashboard layout is quick to understand	The iPhone designs don’t seem easy to use	Work more on the iPhone design
Leo	Its neatly laid out which makes it easy to understand	Some pages come across repetitive	More stats could make it more interesting
Joe	I like the idea of making my own team	There’s not enough team information	Being able to select a squad would be a nice touch
Callum	It’s nice being able to say your position	The home page is too busy	Allowing the coach to message players could be useful
Will	I think it’s an interesting concept	Not enough player information	A way of seeing where the pitches are

Reflection and analysis on Low Fidelity Design

Although our design is basic, the test group seemed to grasp the concept and take a liking to the idea. They gave vital feedback, both positive and negative as well as potential features they'd like to see in the product. Upon reflection we can see that our designs were too busy and could make use of white space, in addition to this we feel the navbar layout is hindering the design of the dashboard.
In summary the key points from this feedback session have been:

Positives –

- **The design was well interpreted**
- **The users were a fan of the concept itself**
- **Dashboard and stats were an element the users enjoyed**

Negatives –

- **Some pages were repetitive and too cluttered**
- **The navbar was not liked by the user**
- **Not enough stats on players and teams**

Suggested improvements –

- **A to-do list**
- **Checking the availability of a player**
- **Squad selection for fixtures**
- **Solution for the coaches to message players**
- **Solution to seeing pitches (maps)**

This feedback will be used when continuing on with the development of our project using the iteration model, we will assess the feasibility of the test users suggested improvements and release multiple iterations with these requested features. We will also look at the negative feedback and develop solutions, this should improve customer satisfaction and experience with the product as the software will be more tailored to the user.

High Fidelity Design (Functional Prototype)

Link to try the prototype: https://framer.com/share/RivalsWireframe--2SMboHdbeQGE7ihQtvvH/jrewVxoFN#jrewVxoFN

For a more detailed view please refer to the appendix (L-T)

Feedback on High Fidelity Design

Our main goal of this feedback session was to get a reaction from the user on the design and basic user experience of the product. A lot of time was spent on developing features suggested by the tester group and designing a marketable product with a unique brand.

Using a tool online (Framer, n.d.) we were able to create a high fidelity (somewhat functional) prototype that demonstrated our vision for the final product. The purpose of this prototype was to understand if our design was successful in being intuitive and satisfactory for the user. 'High-fidelity prototypes are assumed to be much more effective in collecting true human performance data (e.g., time to complete a task), and in demonstrating actual products to clients, management, and others.' (M. Walker, September 29–October 4, 2002) The goal of this prototype is to provide a proof of concept that demonstrates functionality to be tried and tested by customers and clients. To achieve our goal, we must consider the following when testing:

- **Test the right people**
  Again, we tested seven people that we believe would fall into our target audience, we thought this would be beneficial as we could get a reaction to the new design and features from a group that had seen previous iterations.

- **Let test participants contribute ideas**
  Feedback is essential to the development and progress of our project therefore the importance of user feedback cannot be stressed enough. We used the same form from the low fidelity feedback, users completed this throughout testing the high-fidelity design.

- **Have a moderator**
  As this prototype was more complex, we thought it was important to have a moderator to keep the session on-track and guides the test participants through the tasks.

- **Remain Open-Minded**
  On our testing feedback sheets we implanted a "I Like, I Wish, What If" method to allow participants to share their feedback in a critical but positive manner. Also, we made sure not to respond to negative feedback in a defensive manner but instead to ask useful questions to understand what exactly is wrong with our design, and how we can improve it.

- **Communicate with the test user**
  We made sure to inform the tester that the high-fidelity design was functional however would still lack many features expected of the final product, we wanted to see if we had catered to the wants of the user and whether we had been successful with our updated product.

Tester	Keep Doing +	Improve -	Flight
Jake	The colour scheme is aesthetically pleasing	Some pages have a lot going on	
John	The new tool bar on the left of the screen makes navigation easier	There are no iPhone designs	iPhone compatibility
Evan	I think the find a challenge section is unique and intuitive	Would like to see teams addresses instead of distance	Comparing players stats would be a fun feature
Leo	There's a range of stats for both team and player which I like	Some pages come across repetitive	
Joe	Seeing what players are fit or unfit for a game is a great feature	The sign in/up form is abit buggy	A way of accessing the team sheet to see positions
Callum	Nice touch being able to see who says their available for a game	The home page is a bit messy	
Will	Interesting way to challenge teams to games	Not a clear way of picking a team for a match	A map that shows teams and pitches near you

Reflection and analysis on High Fidelity Design

Again, our test group seemed enjoyed the functional concept. They were pleased at the iterations we had made to the product and surprised at the overall quality of the prototype. Superb feedback was given by the group, also bringing some challenging suggestions to the table. Through reflection we can see that our designs have greatly improved however they were still elements of clutter/busy pages. The new features were well received indicating our feedback was authentic, leading to genuine development of the product.
In summary the key points from this feedback session have been:

Positives –

- **The design was aesthetically pleasing**
- **The new tool bar on the left of the screen makes navigation easier**
- **New elements such as the challenge section were unique and intuitive**
- **Plethora of stats for both teams and players**
- **Implementation of a wanted feature (Match fit & Availability)**

Negatives –

- **Some pages are too cluttered**
- **Lack of Iphone designs**
- **Uders want to see exact location of pitches rather than distance**
- **Bugs on Sign In/Up form**
- **No easy way of picking a team**

Suggested improvements –

- **iPhone compatibility**
- **Ability to compare players stats would be a fun feature**
- **A map that shows teams and pitches near you Solution to seeing pitches (maps)**
- **Solution for accessing the team sheet to see positions**

Considering the types of technology to build the final product

To build our product we will use a number of coding languages, frameworks and database management systems. The languages we will most likely use are HTML, CSS, JavaScript, Java, SQL & PHP. Below we have shown the research behind deciding on what stack we will use for building our product.

Front-end frameworks + Stacks

Framework	Popularity	Pros	Cons	Stack
React	80%	• Easy to Learn and Use • Creating Dynamic Web Applications Becomes Easier • SEO Friendly	• The high pace of development • Poor Documentation • View Part	• M - Mongo D • E - Express • R - React • N - Node.js
Angular	56%	• Great selection of third-party integrations • Framework is designed to be fully customizable	• Performance • Steep learning curve • Limited SEO options	• M - Mongo DB • E – Express • A – Angular • N - NodeJS
Vue.js	49%	• Performance • Easy to learn • Concise documentation	• Risk of over flexibility • Language barrier (Chinese)	• M - MongoDB • E - Express • V – Vue.js • N – Node.js

(Front-end frameworks, n.d.)

Database Management Systems

Database	Pros	Cons
MySQL	• Can be used in different OS (Windows, Linux, MacOS) • Data security- MySQL provides a very high-level of security. • High performance-MySQL provides very high performance	• Poor User Interfave • Lack of advanced features compared to PostgreSQL
PostreSql	• Highly expandable • Reliable, stable and secure system • Possible to process complex data types	• Performance degradation in the database environment
MariaDB	• SQL - Is well known and supports most types of usage cases for a database.	• Poor performance on large data sets

	• Open source - support and resources to develop on. • Multi-platform - runs on any operating system	• Caching performance is lackluster
MongoDB	• Performance • Simplicity • Scalability	• Joining in MongoDB can be a tedious task • Duplicates • High Memory Usage

Technology to Build the Final Product Evaluation

We want our final product to have a sleek user experience and with high functionality. To achieve this, we believe that the MERN stack (MongoDB, Express, React, Node.js) will be the best use of technology to build our product.

The decision behind using MongoDB opposed to MySql was because if data cannot be represented in the form of a table, then a fixed table structure is no use, however MongoDB has features to deal with this issue such as, horizontal scalability to balance the increasing load of modern applications.

Express is a server-side framework that helps to develop the web and mobile application using. Express was an easy decision as it's easy to connect with databases such as MongoDB. In addition to this, Express makes Node.js web application development fast and easy. Express improves the performance of our application and works as the middleman between the backend and frontend of the product.

We went with React as it's easy to learn, as we must stay within a timeframe this was essential as we needed to use a framework we can pick up and use without falling behind. Also, creating dynamic web applications are easier with React as it is a JavaScript library for creating user interfaces that may be used to create single-webpage or mobile apps. SEO optimisation will play a big role towards the final deployment of the product, React is proven for better SEO performance as it is a library for building UI, It doesn't need additional tools for SEO such as PhantomJS to tackle SEO issues.

For the backend we decided to use Node.js due to its asynchronous language making performance impressive compared to the average of other languages. Having a high-performance backend means that users of our application will have access to their data much faster. This is vital to the success of the product as the average acceptable time for a user to wait for a web page to load is less than 1 second.

11) A critical evaluation of your concept, your project in its current state and the proposed software project.

References

(n.d.). Retrieved from https://wireframepro.mockflow.com/.
(n.d.). Retrieved from https://www.framer.com/

(n.d.). Retrieved from https://www.framer.com/

Development Life Cycles. (n.d.). Retrieved from learnlearn.com: https://learnlearn.uk/alevelcs/development-life-cycles/

KRIS. (2018, April 30). *Software Prototyping Techniques You Need to Know.* Retrieved from prototypeinfo.com: https://prototypeinfo.com/software-prototyping-techniques-need-know/

Mishra, S. (2019, March 18). *The Importance of Prototyping in Designing.* Retrieved from UXDesign: https://uxdesign.cc/importance-of-prototyping-in-designing-7287c7035a0d

Pal, S. K. (2021, August 11). *Software Engineering* | *Spiral Model*. Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/software-engineering-spiral-model/

Peterheria, N. (2018, January 30). *Why Agile SDLC Model Is The Best for Your Startup*. Retrieved from Product Tribe: https://producttribe.com/project-management/agile-sdlc-guide

S, S. (2017). *A Study of Software Development Life Cycle.*

User Centered Design. (n.d.). Retrieved from Interaction Design Foundation: https://www.interaction-design.org/literature/topics/user-centered-design

What is Iterative Model? (2019, April 30). Retrieved from professionalqa.com: https://www.professionalqa.com/iterative-model

Appendices

Appendix A: User Story Map

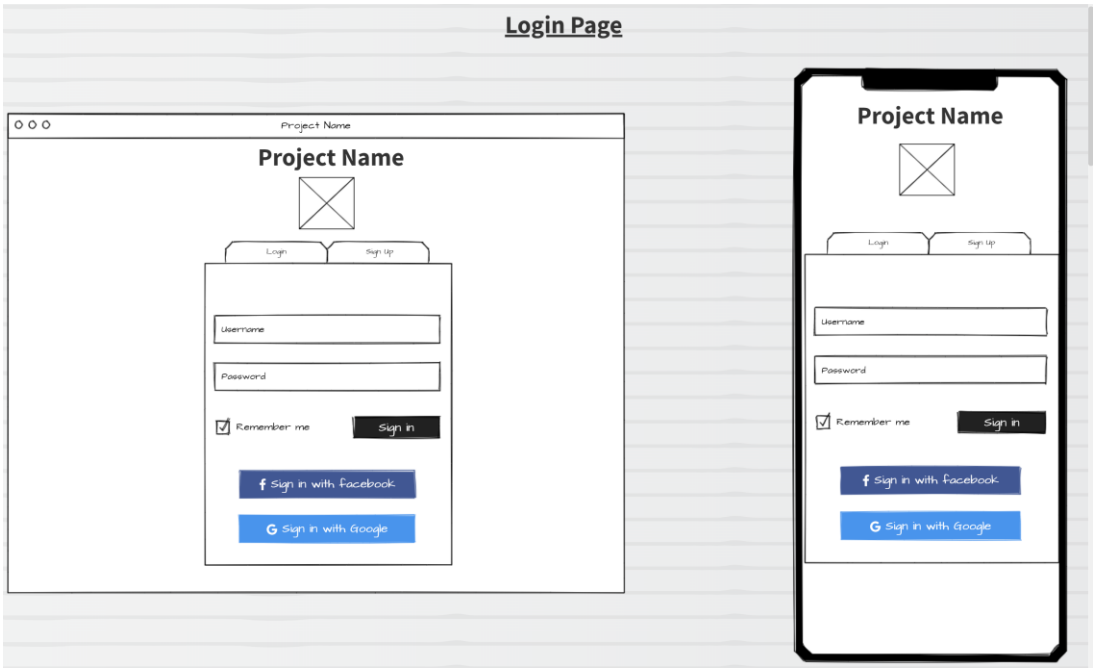
Appendix B: Coach/Manager User Map

User Story Map

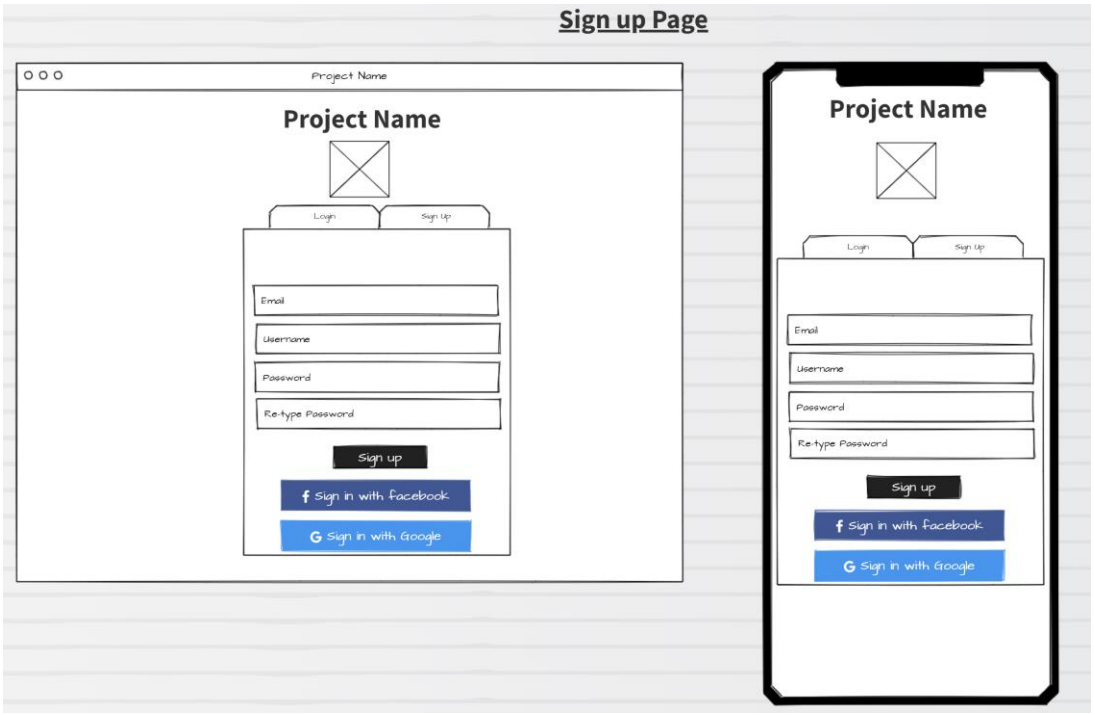
🛒 Football League App User (Manager/Coach)

🏀 **User Activities**	Manage/View Players	communicate with Players or other Teams	Set up a challenge match	create a Team Sheet
👟 **User Steps**	Access either the 'Dashboard' or 'My Team' section	Access the 'My Team', 'Manage Fixtures' or 'Dashboard'	Access 'Find a challenge'	Access 'Manage Fixtures'
📒 **User Stories**	Search by player name	Use the to do list to post things for players to do or remember	Search for oppenent either through swipe or search	Select players eligible to play
	View players who are fit or unfit to play	Message players directly	Select opponent and chanllnege them to a match	Recieve notification when a player confirms availability
	Autocomplete search entry	Message Team coaches/ Managers		Receive notification when player gets an injury and cannot player
				Post team sheet

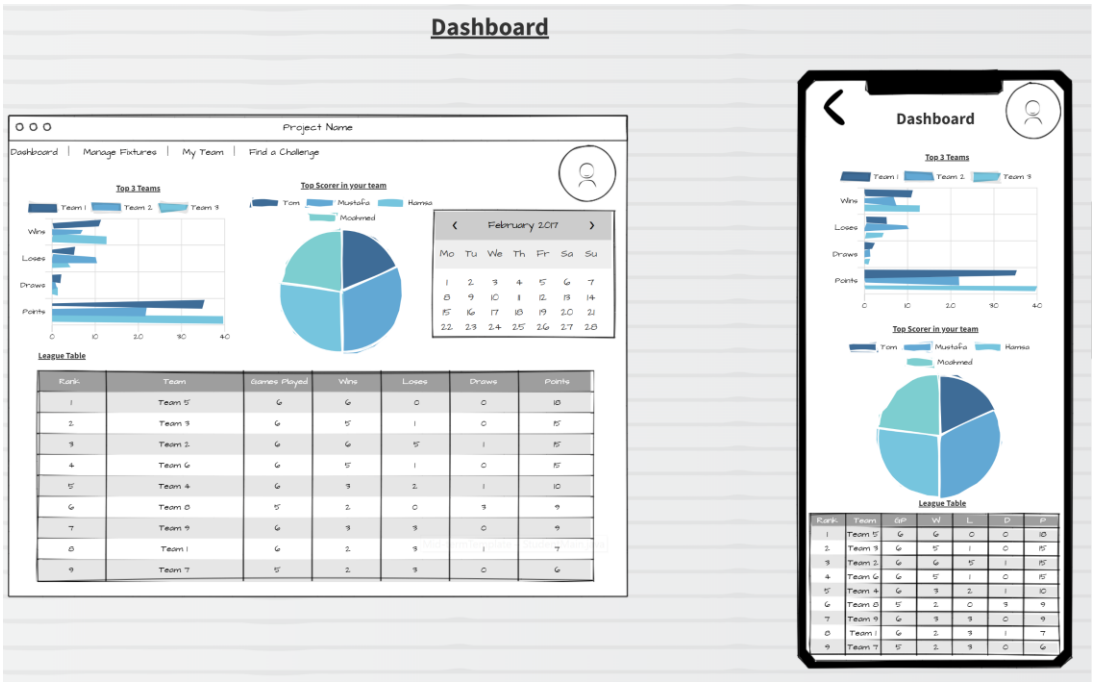
Appendix C:



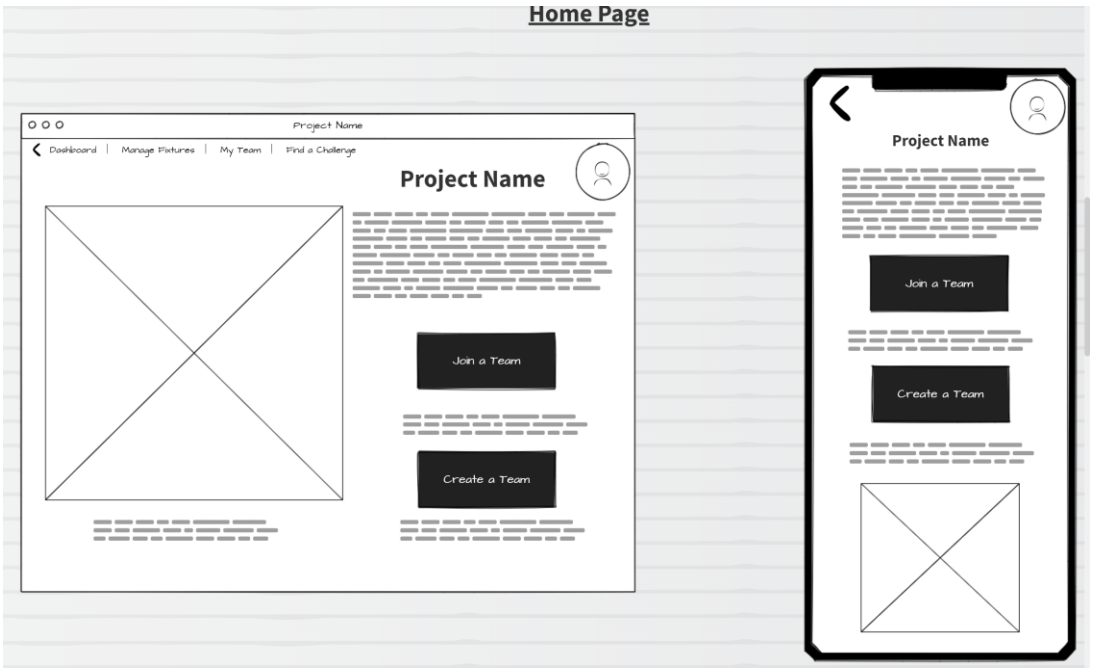
Appendix D:



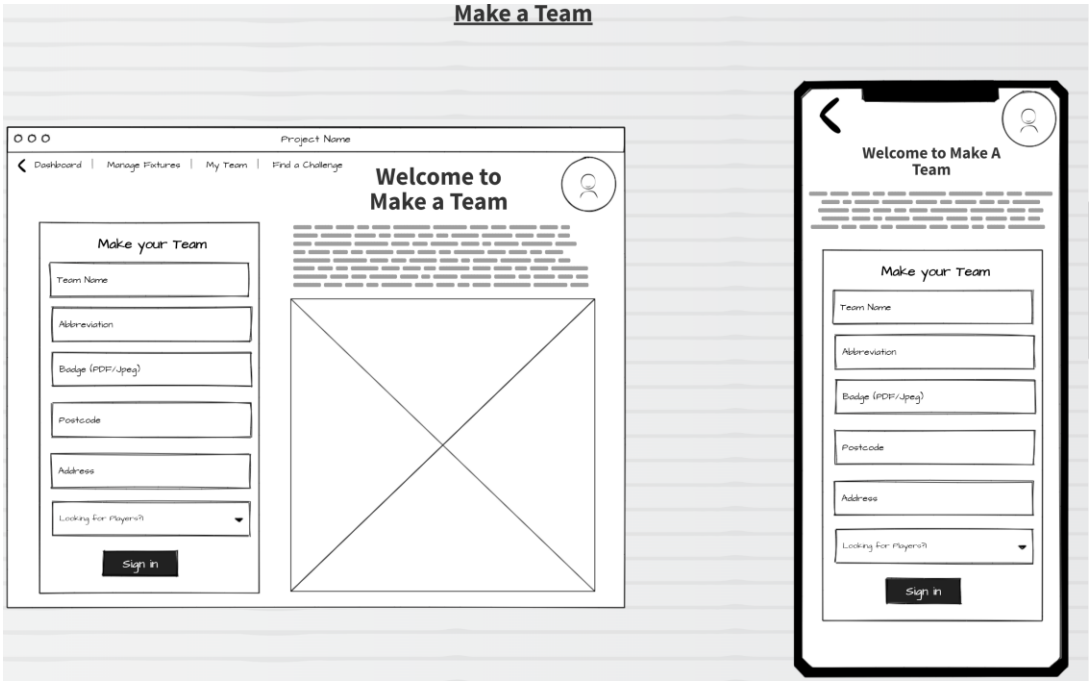
Appendix E:



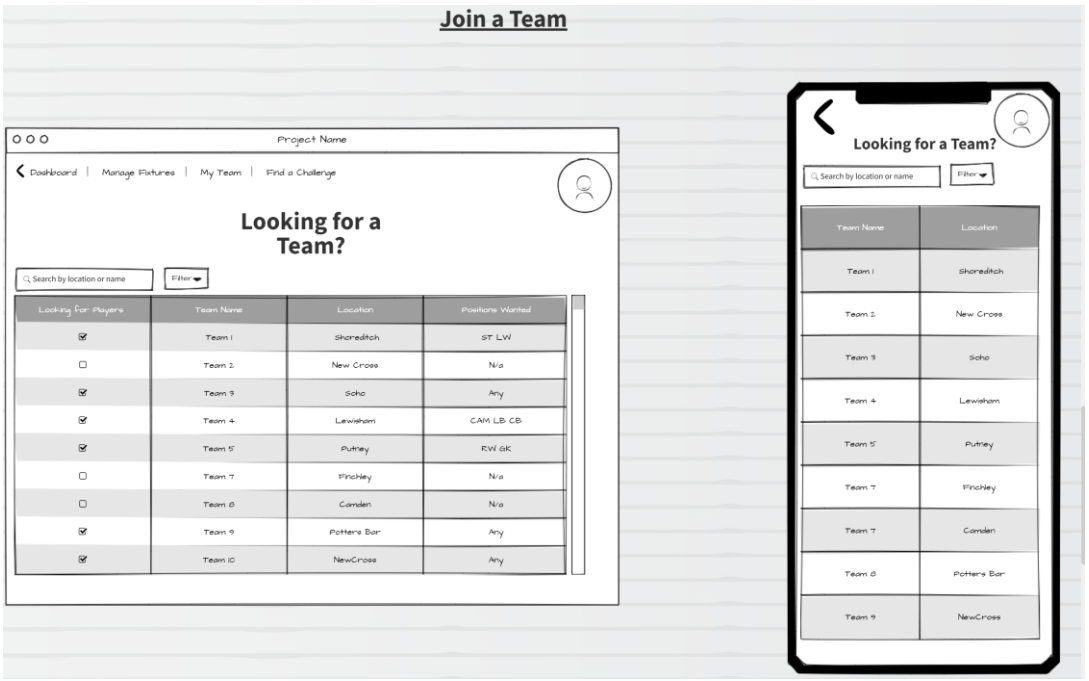
Appendix F:



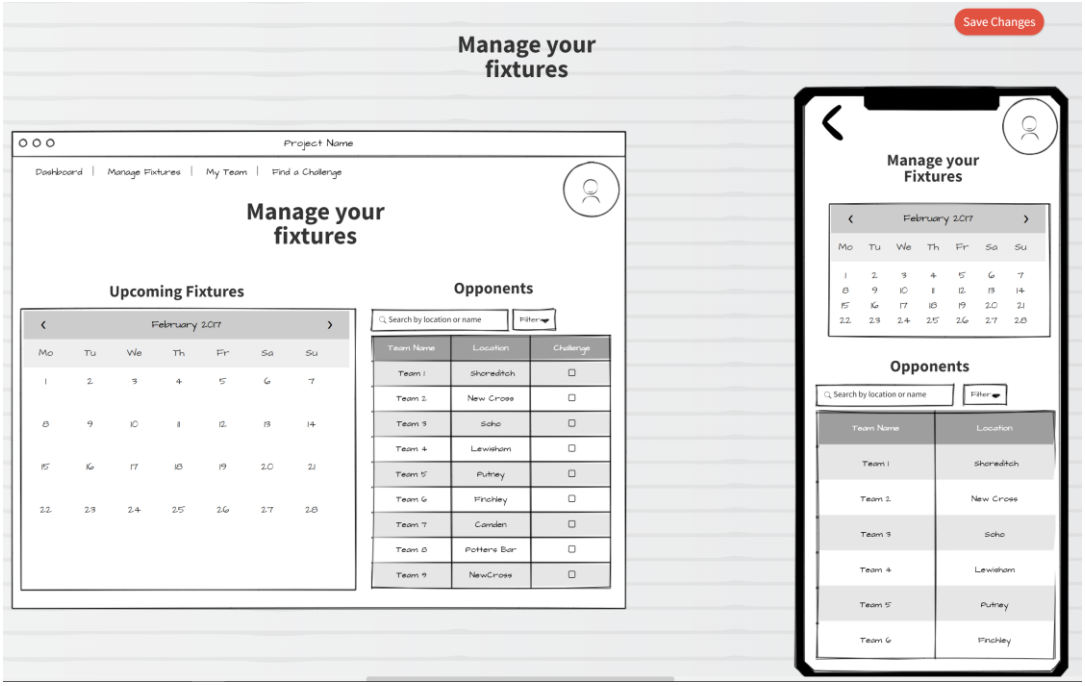
Appendix G:



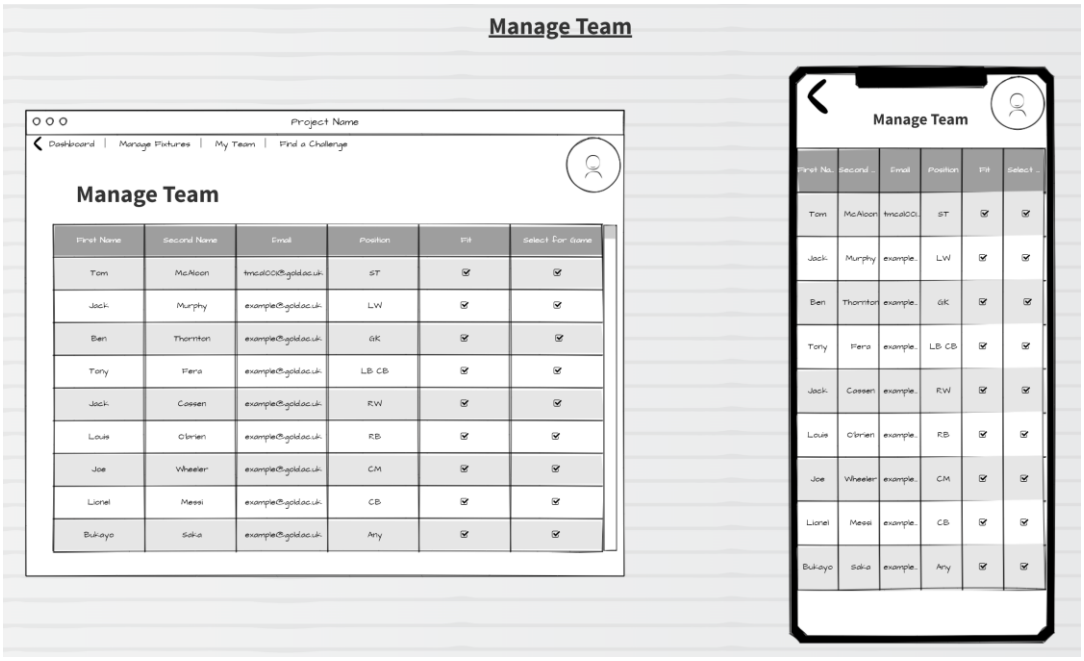
Appendix H:



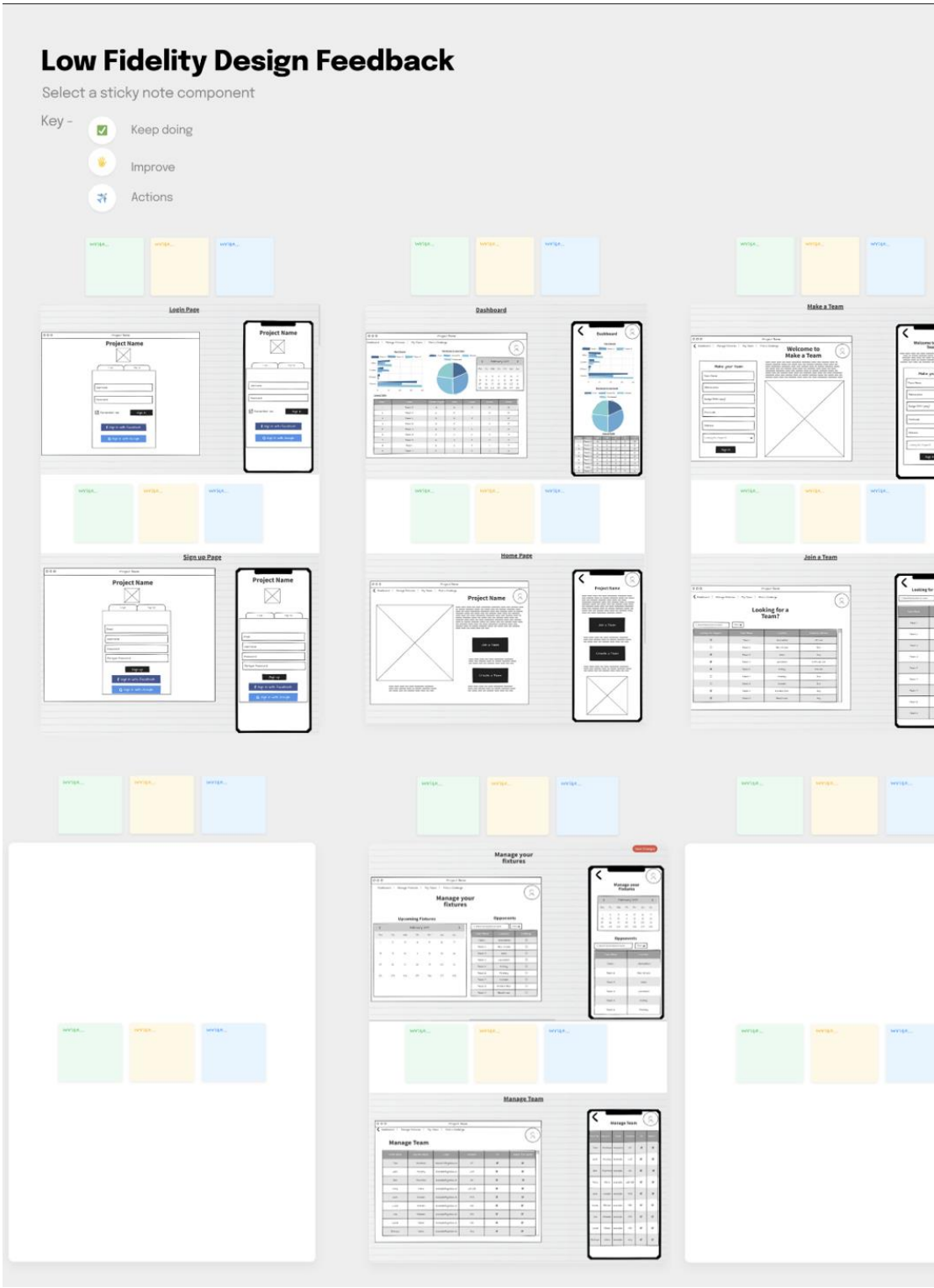
Appendix I:



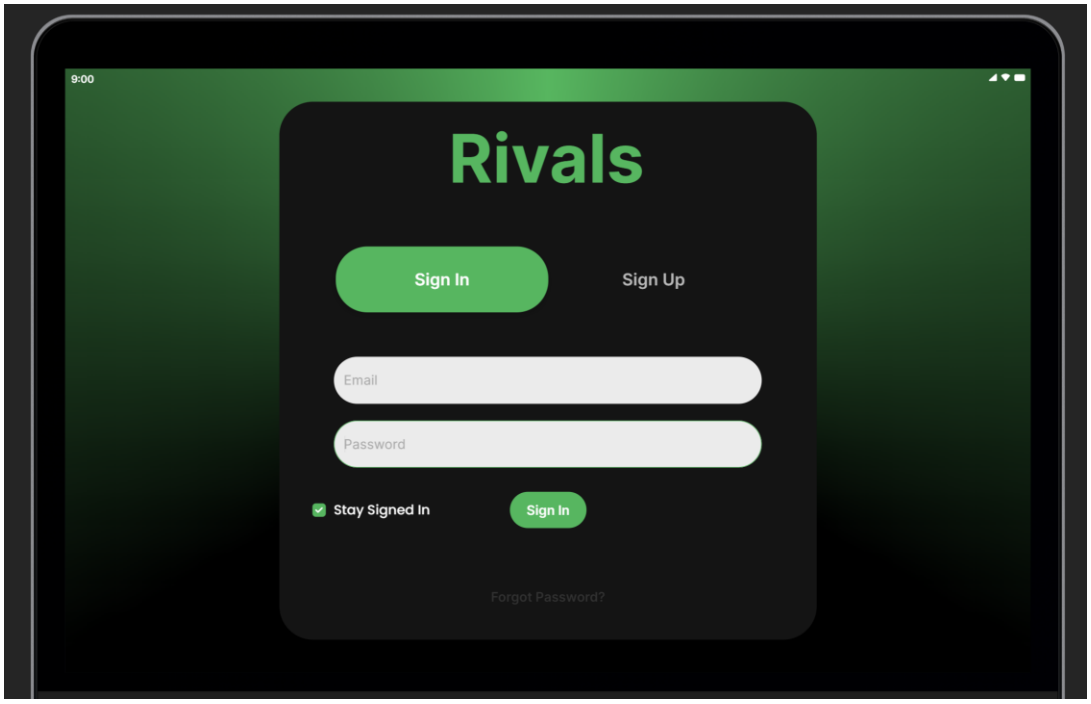
Appendix J:



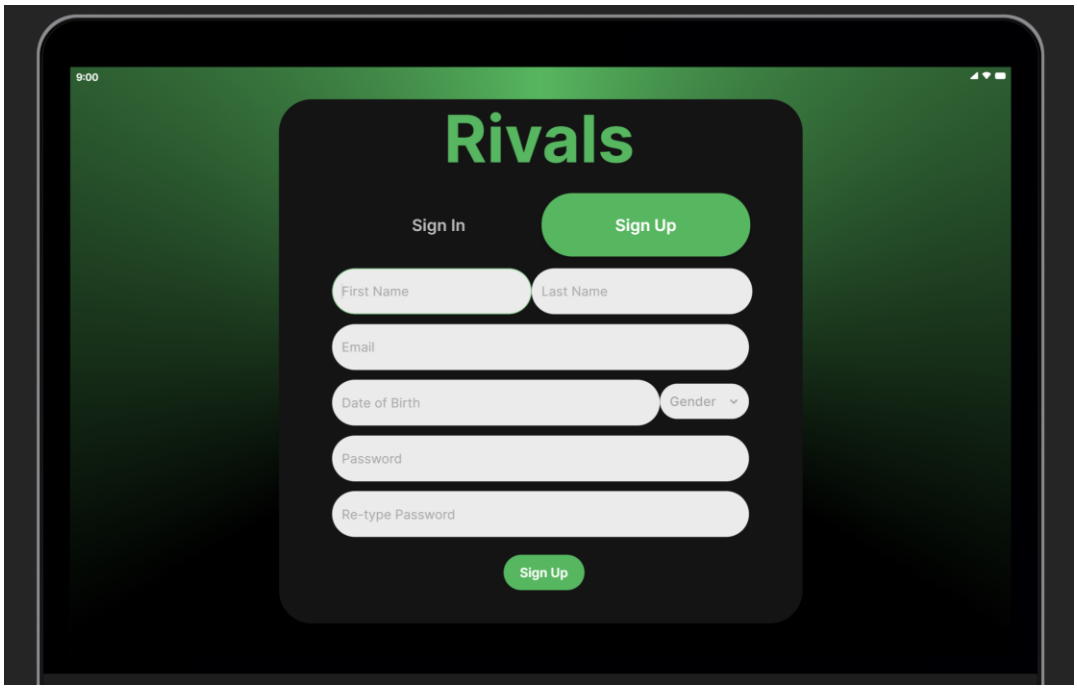
Appendix K:


Low Fidelity Design Feedback
Select a sticky note component
Key -
Keep doing
Improve
Actions
Login Page
Project Name
Dashboard
Make a Team
Welcome to Make a Team
Sign up Page
Home Page
Join a Team
Looking for a Team?
Manage your fixtures
Manage Team

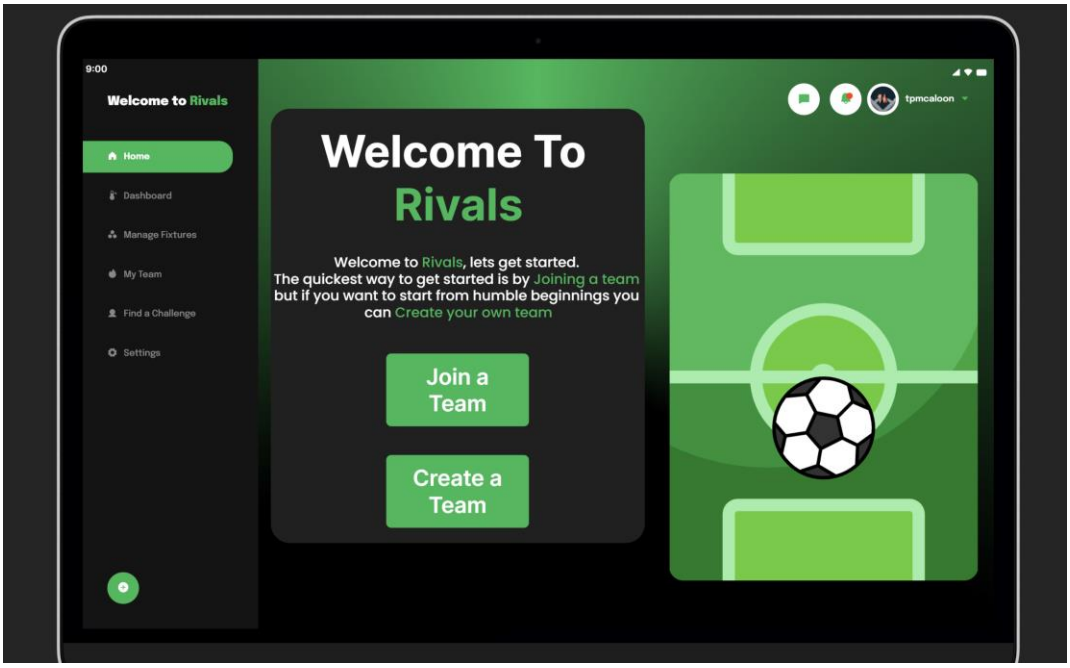
Appendix L:



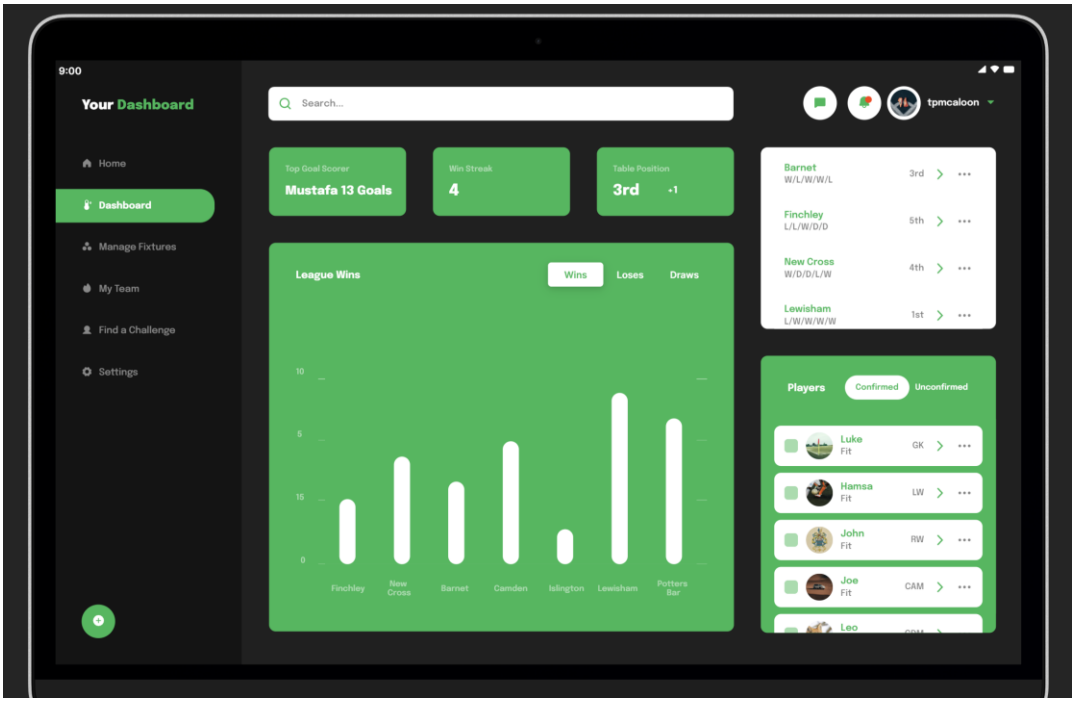
Appendix M:



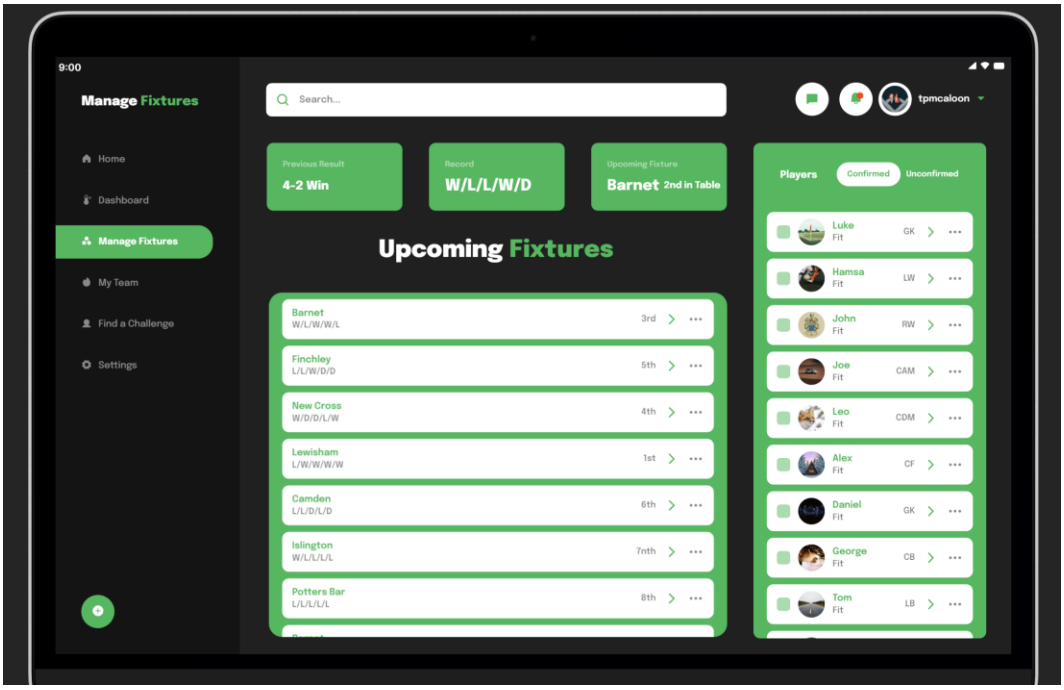
Appendix N:



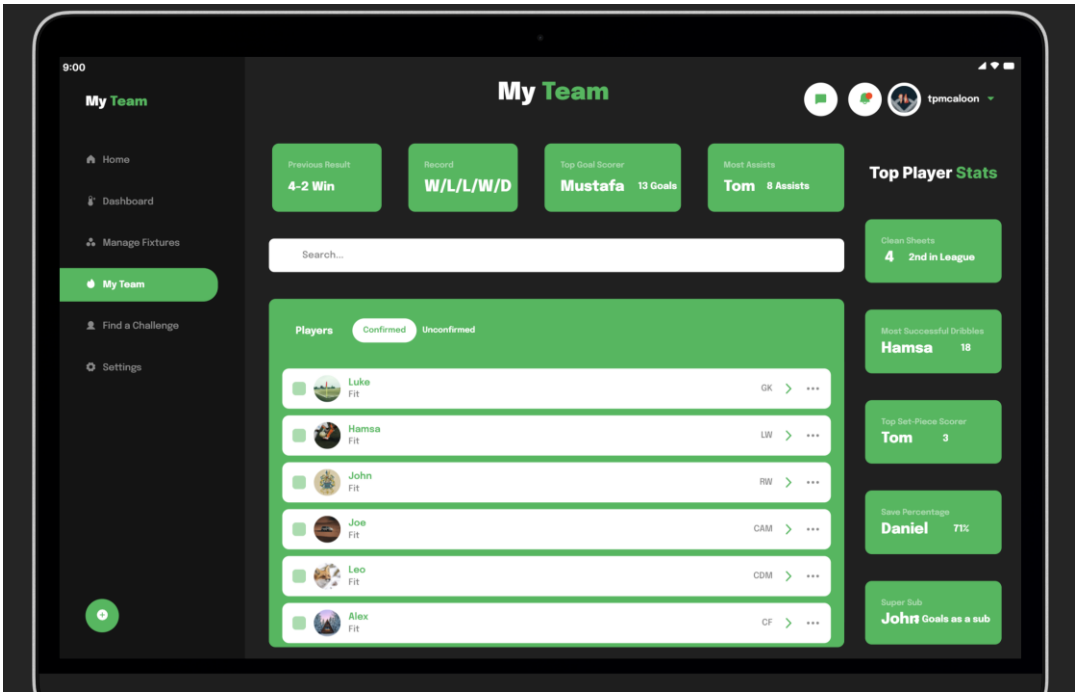
Appendix O:



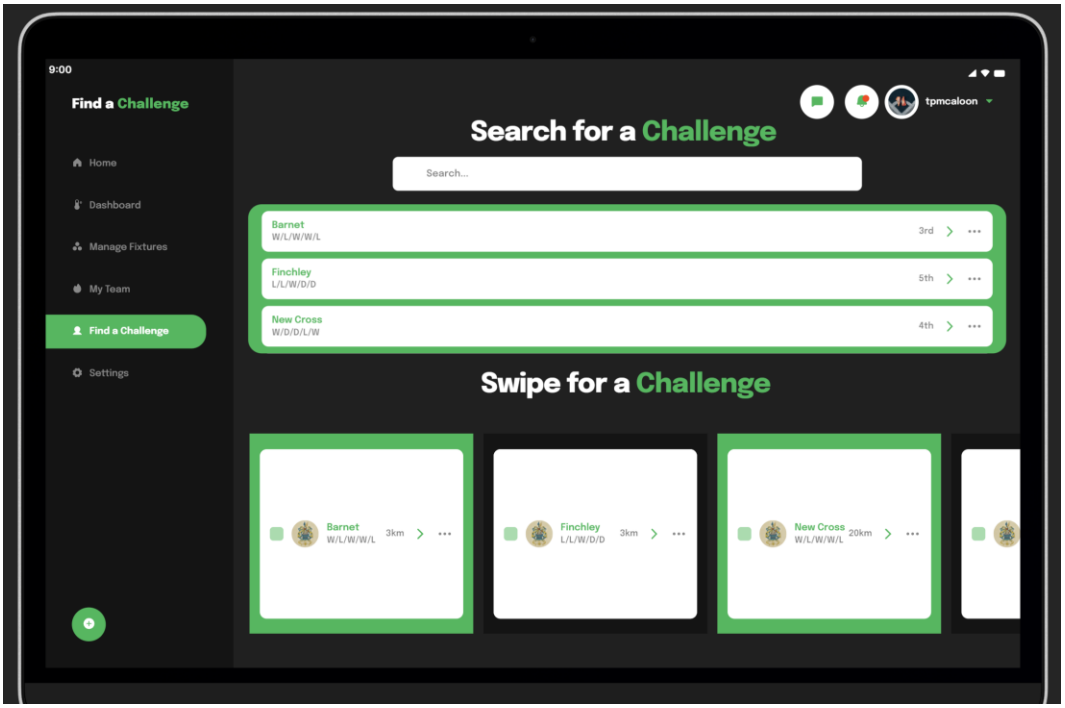
Appendix P:



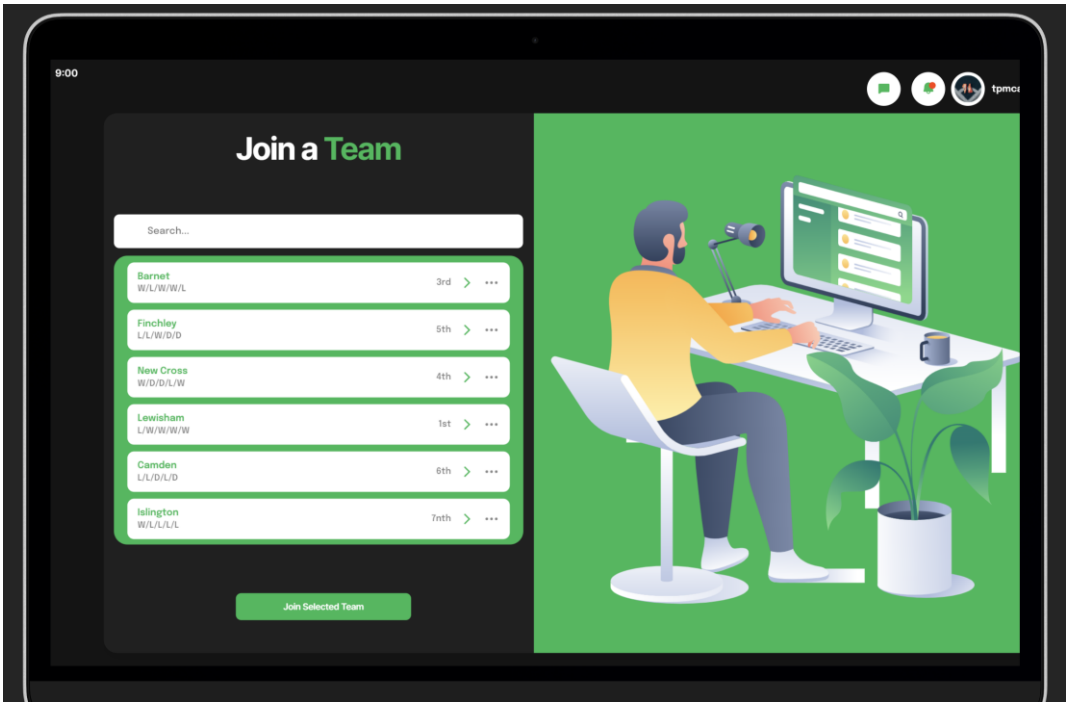
Appendix Q:



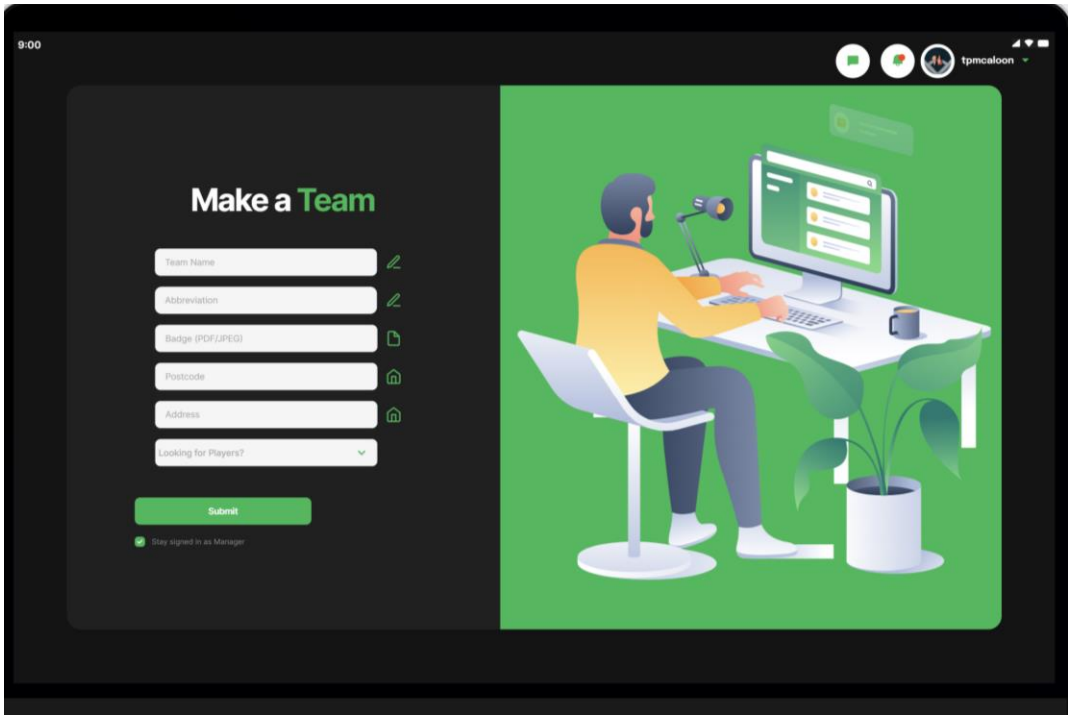
Appendix R:



Appendix S:



Appendix T:


9:00
tpmcaloon
Make a Team
Team Name
Abbreviation
Badge (PDF/JPEG)
Postcode
Address
Looking for Players?
Submit
Stay signed in as Manager