Glossary

* Players - Professional football players from the clubs of the English Premier League
* Users - Users of the site who will sign up to play this game
* Transfer out - A player getting released from a user’s team into the ‘in transit’ area
* Transfer in - A player being brought to replace a vacant spot in the squad due to a transfer out
* Season - Collection of 38 gameweeks over the season
* Gameweek - Collection of a set number of games during a scoring week.

Functionality - This is what we aim to get out of the website. I have my ideas on how to go about doing this from designing and coding the website to make it simple but effective, but if you (the designer) have a better way of implementing the same functionality or creative ideas to make it better, feel free to put forward your suggestions:

1. Ability for players to signup to the website in a simple manner (e.g. username as email address, and alpha numerical password- minimal details required)
2. Ability for the website to withstand large amounts of traffic and do so quickly - on gameday, if all goes to plan, we will have anywhere between 10k to 200k users logging in in a short span of time. i.e. - Quickly loading websites will be a big boost
3. We have seen some fantasy games whereby substituting players on and off requires the page to reload. We do NOT want this. It needs to be interactive and needs to be able to update without reloading the whole page. This may sound unclear but a similar (and more familiar) example will be gmail’s quick reply feature whereby the reply panel opens up without the whole page reloading. Please ask if this is not clear.
4. Back-end must have the ability for admins to set fixtures, gameweeks (move them from week to week for example) etc. Points scoring in fantasy football is based on statistics. At this point in time, we are unsure whether we will have an external source for these statistics (in which case you will need to code for an external xml source feeding into the site in real-time) or whether the admins will manually input the data via the back-end.
5. Players will be grouped in mini leagues of up to 10 teams each (this maximum number of teams should be flexible and modifiable via the backend). Users should be able to generate a league (with a league identifying code) that other users can input in order to join that league.
6. For those players who want to play against random strangers, they are put in a pool whereby teams of ten are generated.
7. One user per league will be the ‘admin’ (either the user who creates the league or someone randomly assigned by the system)
8. Player selection procedure is important (read below for further details)

* Fantasy game similar to<http://fantasy.premierleague.com> but with some differences:
  + Users play in small leagues of up to 10 teams (i.e. 10 users per league)
  + Each team has a squad of 15 players, 2 goalkeepers, 5 defenders, 5 midfielders, 3 forwards
  + Each player can only be chosen by one user in a league. E.g. If Wayne Rooney has been chosen by user 1, no other user in the league can have him in their team. i.e. - This is to prevent identical teams and make the experience more realistic and enjoyable
* Advertising model
  + Adsense capability
  + Pay per click capability
  + Pay per impression capability

Components

1. User registration/league registration
   * One team per user
   * Each team can only be a part of one league
   * League registration
     + A user should have the capacity to create a league, which will generate a code that can be passed onto other users, who can in turn enter this code
     + The user that creates the league becomes the league ‘administrator’
   * Random league generator
     + For users that want to join random leagues with strangers (don’t have enough people to make their own league)
2. Player selection procedure
   * Administrator sets off selection
   * Once selection procedure begins, no more entries to the league can be made
   * The system will randomly sort users in an order
   * Selection will begin and continue one by one via snake method: The order reverses every round. e.g. In a league of 5 players, round 1: 1, 2, 3, 4, 5; round 2: 5, 4, 3, 2, 1; round 3: 1, 2, 3, 4, 5; until all 15 rounds of the selection are over:

* + - * "serpentine" or "snake" draft, owners take turns drafting players in a "serpentine" method, i.e. the owner who picks 1st in the odd rounds picks last in the even rounds, in the interests of fairness.

* + The system must have a ‘queue’ method - Since users may be all over the world, with time differences etc. and there may be time before their turn comes again, to speed up the process, the queue will allow users to tell the system the order of their next preferences, so that the system can auto-select for them when it is their turn, speeding up the process.
* Player transfers/trades - (Just ideas here, the aim is to allow players to make trades, if you, as the designer have a better idea on how to do this, feel free to contribute)
  + Players market status will fall under 3 sections
    - Under contract - Player has been selected and is currently in the team of another user. Player is not available for selection
    - Free agent - Player is not under contract (in any team), nor is he ‘in transit’ (see below). Free agents can be transferred into a user’s team freely
    - In transit - KEY area. Serves 2 functions:
      * When a player is transferred out, they move to the ‘in transit’ area, stay there for 48 hours before they move into the ‘free agents’ list (system needs to log the timestamp when a player is transferred out and automatically move players to the free agents list when the appropriate time arrives). The purpose of this is to allow users across different time zones and users that are busy to get a chance to see which players will become free agents soon
      * Allows player trades: If users want to make a trade, this area will serve as a ‘go between’. The way I envision this will work (open to suggestions) will be by allowing the previous owner of a player to nominate a fellow user to take up this player before he becomes a free agent and vice versa.

Example one - If user 1 owns Rooney\* and user 2 owns Aguero\* and they come to an agreement to do a player trade, they can both do a ‘transfer out’ and nominate each other. i.e. user 1 nominates user 2 as the next owner of Rooney, allowing user 2 to ‘transfer in’ Rooney while he is still ‘in transit’ and vice versa. This example is a simple one that doesn’t particularly require an ‘in transit’ area to happen, but I feel it will be useful for more complex player trades.

\*Examples of real football players

Example two - More complex trades will occur when more than one position is part of the trade OR more than two users are part of the trade. If user 1 wants to trade Rooney\* for user 2’s Silva,\* then a straight swap is not possible as Rooney is a fwd and Silva is a mid. In this scenario we are assuming that the replacement for Silva is a mid in the free agents that user 2 has in mind while the replacement for Rooney is a fwd that is also a free agent.

Example three - User 1 wants to get rid of player X for player Y, user 2 wants to get rid of player Y for player Z and user 3 wants to get rid of player Z for player X, then (after all users have agreed on the trade) user 1 will release player X nominating user 2 to take him up, user 2 will do the same for player Y (nominating user 3) and user 3 will do the same for player Z (nominating user 1), allowing a 3-way trade and 48 hours for the trade to occur.

Essentially, for player trades to happen smoothly, the following components are required:

-Nominating fellow users to take up a player you have released

-Stopping players that have been released from being taken up by other (non-nominated) users

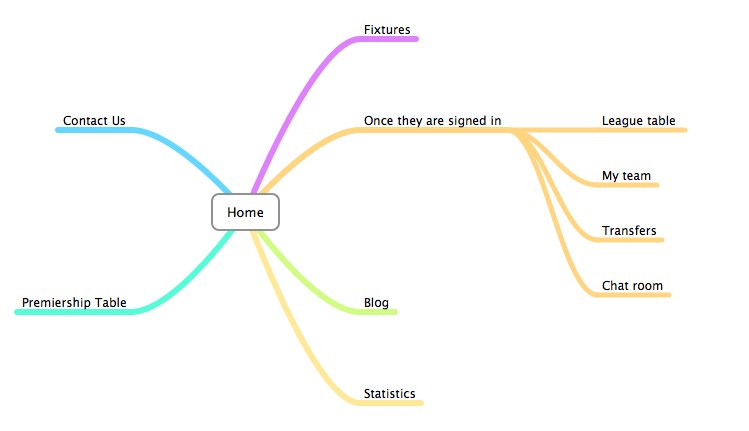
* + A player can be sold at any time. There should be a ‘transfer’ page for this functionality. When a player is sold, he is moved to the ‘in transit’ section.
  + A transfer can only be completed when both the ‘player in’ and ‘player out’ components are satisfied.
  + Players can only be bought if they are a ‘free agent’ or if they are the nominated user if they are ‘in transit’
  + Players, once NB -
* Database
  + Players
    - Fixed info
      * Position (GK, Def, Mid or Fwd)
      * Club - Backend requires an area that allows for setting the 20 teams each season (due to relegation and promotions from season to season changing this)
      * Club 21 will be ‘ghost’ - A big part of draft fantasy is selecting transfer targets that have not yet been bought by their target clubs. In some cases, users may want to select a player that is not yet a part of the premier league. This will allow us to add a player and have him unclassified and up for selection.

E.g. If Manchester United are rumoured to buy Ronaldo, then Ronaldo will be added to the game and classified under a ‘ghost team’. This will allow users to choose Ronaldo/transfer him in even though he is not yet in the premier league. The cost of doing this, of course, is the possibility that the player does not end up joining the premier league in which case the opportunity cost has not paid off.

* Points info. Players will be awarded points for actions they perform (e.g. scoring goals). As mentioned earlier, the stats that power this may be from an external source or from the admin (to be confirmed). In any case, the back-end must have an area that specifies the number of points awarded to each player for each action depending on their position.
  + E.g. A goal for a forward will be 4 points. So the backend must have a section whereby we can specify ‘4’ as the number of points for forwards when it comes to scoring goals.
  + The same will apply to midfielders, defenders etc.
  + The same will apply to other actions: e.g. negative points for yellow/red cards, assists etc.

* Rules for starting 11 for each gameweek (To be discussed in further detail over Skype so that understanding is fully clear):
  + Formation chosen must have exactly 1 GK
  + Formation chosen must have at least 3 defenders: i.e- 3-4-3, 3-5-2, 4-4-2, 4-3-3, 4-5-1, 5-4-1, 5-3-2 or 5-2-3
    - * Substitution order matters (just like [fpl](http://fantasy.premierleague.com))
      * System should have autosubstitution capability based on substitution order
      * Captain and vice-captain chosen each week
      * Captain’s points are doubled
      * Vice-captain only comes into play in the scenario whereby a user’s captain does not play (i.e.- in terms of database, only if the captain’s minutes played, for that gameweek = 0). In this situation, the vice-captain becomes the captain and he will score double points for that gameweek.
    - Market status
      * Free agent
      * Under contract
      * In transit
  + Fixtures and Leagues
    - Each ‘gameweek’ will consist of the games played in real life during that gameweek. The backend will need to have a fixture management system that will allow each gameweek (GW 1-38) to be allocated individual games. This will usually be 10 but may be more or less depending on how fixtures are played out in real life.
    - Each gameweek will have a predetermined ‘deadline’. The backend gameweek management section (whereby fixtures for each gameweek are determined), should have the option to set the deadline for each gameweek. Squads and lineups, at that point in time, for each user in their league, will be locked in and any changes made afterwards will only apply to subsequent gameweeks. The system needs to have it in place to have a window, just after the deadline (as short a window as possible) where the system updates. After the system updates:
      * The latest points page for each user in the league table will show the starting 11 and
* Player chat area
  + A continuous league chat room that allows for banter, negotiations and random chatter - similar to a whatsapp group chat
* Points scoring
  + Back-end should have the capacity for live updates. As an update is applied to the backend to a player, every team in all leagues that has that player *for that gameweek*
  + i.e. - When an obsessive fan goes to check his team right after a goal, an updated points tally should be visible
* Blog
  + Articles to help with SEO
  + Ideally using<http://sett.com/> - more on this later. Ideally, the same username and password should apply.
* Futureproofing IMPORTANT:
  + Iphone app
  + Android app
  + We intend on turning this website into mobile applications. The programming language used, the database built etc. should all be such that apps can be made with the full functionality of the website, in the future..
  + For this reason, you if you/your team have experience developing apps, it will be a big advantage

Front end rough mock-up of site map:

This should be enough information to provide a detailed quote. Feel free to ask any questions if needed. More thorough details about each page and each element will come once project development has begun.