Generating Cricket Fixtures

Summary

This document discusses in some detail the provision of sports fixtures in a multidivisional league and potential tooling to make this exercise less onerous. A prototype solution has been written as proof of concept and has produced promising results.

Although primarily coming from the cricket community many of the problems in creating recreational sports fixtures are common across many sports.

In cricket the main items which need to be accounted for in creating fixtures are

- Integrity of the competition and social requirements meaning fixtures are played on a regular basis; each team plays the same time and day each week.
- Competition fairness means that all teams must play fixtures with a balance between home and away fixtures, and must have reasonable gaps before the reverse fixtures
- Teams share grounds so fixtures need to be scheduled to avoid two or more teams needing the same ground at the same.
- There are then restrictions or requests placed on leagues by clubs who wish to play some matches either away or at home on a specific date.

Often the manual solution to this is to use the same templates to generate fixtures for each division placing teams into numbered slots which then produce a predetermined pattern of fixtures. This pattern guarantees the ground sharing and other requirements are met.

This document discusses an approach to automate this process by taking the constraints and allocating teams to positions in the template which match those constraints.

It is designed to work in a majority of cases noting there are some problems in this area that are impossible to resolve. It can also identify some instances where a problem can not be resolved.

The solution is a productivity tool to allow the fixture administrator to set up the restrictions and then warn or prevent the fixture administrator inadvertently placing a team into the fixtures in a place which breaks those constraints.

It also includes a feature to automatically complete the allocation for all teams. Initial testing this automatic allocation algorithm which is based on a backtracking approach has produced fixtures for complex cricket leagues in timeframes ranging from a few seconds to a few minutes compared to manual methods which can take hours or days.

The prototype solution is stand alone and takes feeds from the ECB Play-Cricket competition management system. The solution could remain standalone or be more tightly coupled into competition management software; both requiring some degree of further development and provision of a support structure.

Introduction to fixtures in cricket

Cricket in England and Wales is organised such that some 6000 cricket clubs play in organised competitions run by about 600.

The leagues organise competitions which fall into two main types.

- Round Robin Leagues in which all play all a set number of times and are awarded points based on performances; the overall winner accumulating most points across all matches.
- Cup competitions were a winner progresses to the next round and a loser either goes into a best loser's competition or is knocked out.

This document in the main deals with league fixtures; in a typical season we estimate around 200000 of these are generated for cricket leagues in the UK.

Whilst this document deals mainly with recreational cricket. Other sports and games have similar requirements to organise league competitions many of which are broadly similar.

There is a distinction in requirements between recreation sport and elite sports. In recreational sports participants generally can only play during leisure time and therefore the time constraints are such that a league will often say 'all matches are played on Saturday at 1pm'. Compare this to elite sport where schedules are more likely to be influenced by media coverage, avoiding excessively short or long periods between matches and avoiding 2 sets of spectators converging in the same place at the same time.

In recreational cricket it is common to have a routine of playing at a certain time each week. The same would be true in other team sports such as football, hockey and rugby. One difference in cricket which often means timings of fixtures are more important than these other team sports is that a cricket fixture if called off due to weather conditions it does not get re-organised whereas other sports try to rearrange such fixtures. This means there in more emphasis in cricket on everyone playing at the same time on the same day to ensure the maximum fairness. Cricket matches also traditionally last longer than most other sports, so whilst in hockey it might be possible for the 1st XI to play in the morning and the 2nd XI in the afternoon on the same ground this is seldom possible in cricket.

The venues are also key. It is likely that grounds will be owned or rented by the clubs and not in direct control of the league; then because of upkeep costs these grounds they need to be used to capacity each week. As a team sport there is the concept of a home team and away team; such that half the fixtures are home fixtures and half away fixtures. 1 team therefore needs a home ground for half their fixtures and 2 teams can share the same ground.

Whereas in professional sport the venues can be generally be guaranteed to be available whenever needed in recreational cricket (and other sports) there will be competing requirements on that availability

Generating the fixtures for any sporting league is part science and part a manual effort to fit all the different requirements from clubs and the league itself.

If the league requirements and structure fall into the simpler categories, the generation of an elegant and balanced set of fixtures can be relatively simple based on the mathematics.

If there are complex requirements, then manual manipulations are usually necessary and compromises made to the elegant solution.

This document looks at the problem and sets out one possible solution and method; it is aimed at both the person sitting there looking at the issue and maybe not even knowing where to start, and those who have some experience through to a discussion on some of the common issues and solutions needed.

It then introduces the possibilities of semi automating or in some cases fully automating the fixture generation process.

Requirements we need to Address in creating fixtures

The basic requirements

At a minimum we start with a number of general requirements:

All play all a set number of times - most commonly but not always twice, once home and once away

The key requirements are

- All grounds used each week
- No ground double booked where a ground is shared
  - Sharing might be within a division, across divisions or even between 2 leagues
- Fixtures where the second half of the season repeats the first with home and away reversed
- Teams do not have long runs on consecutive home and away matches, with no more the 2 home games in a row and not more than 3 in any 5 being the optimum.

In some circumstances, fulfilling just these requirements for every team in the league might be good enough.

The more complex things

It gets more complex when the league or the clubs have some specific requirements for instance

- A specific day where a ground is not free
- An extensive need to match ground sharing with another league
- A need to have divisions with unequal numbers of teams

Here the key is understanding what is possible albeit sometimes difficult and what is simply impossible to do.

A ground is not free

If a club uses a ground for 2 teams for a league playing a match a week across a set number of weeks which is ‘the normal’ situation then there can not be a free date at a ground.

Clubs may look at this and say; simple; reverse a fixture to give a club 2 away games. This may work for that club but does not work overall in leagues were clubs play each other home and away as on the reverse weekend the club will have 2 home matches.

There are some exceptions to this is if there are an odd number of teams in a division where there will always be a bye.

It may be possible then to place the team wanting no home fixtures as playing at home against the ‘bye’. This situation can be engineered by carefully examining the fixture generation templates and placing both the team from the club concerned and the ‘bye’ into the correct places. Doing this whilst possible does take away flexibility from the fixtures admin and is somewhat difficult to sort out.

What is clear is that if multiple clubs have this request then accommodating it even in divisions with odd numbers of teams quickly gets impossible.

Another similar approach to this issue is to arrange the fixtures for the difficult week to be home versus a club which does not share a ground and has exclusive use of that ground every week.

Then by agreement between the clubs the home fixture remains the home fixture but is played at the away teams ground. Some leagues will allow this others will view it as too big a compromise to the integrity of the competition.

This again means ensuring pairs of teams are in the correct positions in the fixtures template.

If all teams in a division share grounds then this immediately becomes impossible.

The other usual resolutions in the cases were a league determine this to be impossible is to place the lower ranked team at home and then

- Move that fixture to a different date
- Allow the home club to find and use an alternate ground usually hiring this from another club, or third party

This leads to a modification of the clubs request to be 'my 1st XI must be away on that date.

Matching fixtures with another league

If a single team in one division shares a ground with a single team in another league then a conversation between the leagues can often determine that both have patterns of fixtures which go HAHAHA and AHAHAH etc. This is because even where leagues use different historical template to fulfil the basic criteria it is likely but not certain that the templates used have these patterns. These are the easy patterns to remember and therefore agree to use rather then looking for other 'might fit' scenarios.

In this 'one off' type case it may well be possible for the leagues to compile fixtures adhering to this simple rule for just that team or a small number of teams. There are however situations where many clubs have teams in 2 leagues, for instance a 1st XI in a premier competition and a 2nd XI in a feeder competition. In these cases it is key that the leagues work together and also use the same templates to generate their fixtures.

It is possible to do one set of fixtures first, usually those in the higher ranked league and use the same fixture patterns lower down to do the lower ranked league later. In doing this there are some potential issues to be aware of.

If a league has divisions of unequal size, matching the fixtures for the first half of the smaller divisions season is usually simple but after that the patterns diverge and some difficult manipulations are needed to preserve the ground sharing.

The league doing fixtures first can also make it easier for the other leagues and reduce then possibility of an impossible problem coming up. This is achieved by for each pair of clubs with teams sharing between leagues use the complimentary slots in the template. This is explained further in the discussion on how the templates work.

Unequal sized divisions

This is a topic in its own right and will be discussed later as it needs an understanding on how fixtures are generated from templates.

Solutions Characteristics

There are several different tactics that can be used to fixture a multidivisional sport leagues with restrictions on match timings and ground sharing to get a workable set of fixtures.

The approach used by many is templates and that is the main consideration of this section. Templates are a good way to solve the generation of fixtures if the basic requirements revolve around ground sharing and playing at the same time each week. Other approaches can work better if the base requirements are different for instance minimisation of travel distances (Home and away in say in major league baseball is not as important as managing the road trips).

Other approaches are valid including trying to do brute force allocations by trial and error. The problem with a true brute force approach is the number of potential combinations which may be needed to find an adequate solution. The number of different ways to combine 10 teams is over 3.5M and that is only 1 small division.

This is why many people use templates, some of which have been passed down through many years of fixtures administrators. What these do is add some symmetry to the problem and therefore cut out some of the random combinations and mean if a solution is achieved, we know it has characteristics which by definition meet the basic requirements.

Even then however when the problem moves from a single division to a multidivisional league the problem becomes harder to solve and can involve hours with pen paper or spreadsheet. When solved a minor change such as 1 team dropping out to be replaced by an alternate can cause a ripple effect meaning that we have to start again from scratch. As these things often happen last minute then when teams do drop out administrations can be forced to leave gaps in the fixture list.

Template to address the basic requirements - single division?

This is a lay person's description of the mathematics of how these templates work

https://en.wikipedia.org/wiki/Round-robin_tournament describes a round robin tournament. A chess grand master Johan Berger devised a method to generate elegant templates for such round robin tournaments which have since been adapted by others.

These templates generate fixtures in a reliable and consistent way. If you start with a pen and paper and say team 1 plays team 2, Team 3 plays Team 4, Team 5 play Team 6 in week 1 and try to construct these with no method you will may well fail especially as the number of teams increases.

There is however a simple way to generate these templates.

A basic method, illustrated here with a 6 team example, is start with '1' in the top left corner and rotate the other number, 1 place clockwise each week and you get all play all fixtures albeit if you left as is team 1 would be at home all 5 matches!

Week 1			Week 2			Week 3			Week 4			Week 5	
1	2		1	3		1	5		1	6		1	4
3	4		5	2		6	3		4	5		2	6
5	6		6	4		4	2		2	3		3	5

Fortunately, there are already people who have adapted the basic method, reversed a few fixtures and come up with really elegant fixture templates, for instance this for a 6 team league

Week 1			Week 2			Week 3			Week 4			Week 5	
6	5		3	6		2	6		6	4		1	6
2	3		5	1		4	5		1	2		3	4
1	4		4	2		3	1		5	3		2	5

With a bit of study, you will see no team is at home more than 2 times in a row, but you also might spot every time team 1 is at home team 4 is away, same for team 2 and 5, and 3 and 6. You might also notice Team 2 plays Home, Away, Home, Away, Home and Team 5 therefore has the opposite pattern Away. Home, Away, Home, Away.

There is are alternative ways of presenting this pattern with one such way being

It is sometimes easier to see this visually if you rearrange this into a grid so.

	Week 1	Week 2	Week 3	Week 4	Week 5
1	4			2	6
2	2		6		5
3		6	1		4
4		2	5		
5		1		3	
6	5			4	

Here in orange on the left is the Home team, the Weeks are across the top and in the grid are the away teams. In week 1 Team 3 does not have a home fixture so has blank against it. Team 6 is hosting Team 5 so has a 5 against it.

With this presentation some will find it be easier to spot the patterns where 1 is always away when 4 is at home, 2 and 5 and 3 and 6 then match in the same way.

So now if you had a single 6 team league with all play all you can repeat the pattern but reversing home and away in weeks 6 thru 10.

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10
1	4			2	6		5	3		
2	2		6		5		4		1	
3		6	1		4	2			5	
4		2	5			1			6	3
5		1		3		6		4		2
6	5			4			3	2		1

Which is the same as

Week 1			Week 2			Week 3			Week 4			Week 5	
6	5		3	6		2	6		6	4		1	6
2	3		5	1		4	5		1	2		3	4
1	4		4	2		3	1		5	3		2	5
Week 6			Week 7			Week 8			Week 9			Week 10	
5	6		6	3		6	2		4	6		6	1
3	2		1	5		5	4		2	1		4	3
4	1		2	4		1	3		3	5		5	2

Therefore in our simple example if we now wanted to do fixtures for our league and say Team A shared with Team B, and Team C with Team D, and Team E did not have a ground available on Week 1 or Week 10, we can use the patterns to:

Place Team E into Slot 3 as this has a blank = an away fixture in Week1 and Week10. Then place Team A in Slot 1 and Team B in Slot 4, Team C in Slot 2 and Team D in Slot 5. Which leaves Team F in Slot 6 and everyone’s needs are met.

Although the example here is for a 6 team league the method is a general one and can be applied to any number of teams in a division and generates n-1 weeks of fixtures where n is the number of teams.

Here for example is a 10 team pattern. It is not the only solution and other solutions are just as good. It has the attributes

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	6		8		10	2		4			7		9			3		5
2	5		7		9		10	3			6		8		1			4
3	4		6		8		1		10		5		7		9		2	
4		10	5		7		9		2	3			6		8		1	
5		3		10	6		8		1	2		4			7		9	
6		2		4			7		9	1		3		5	10		8	
7		1		3		5			8	9		2		4		6	10	
8		9		2		4		6		10		1		3		5		7
9	7			1		3		5			8	10		2		4		6
10	8		9			6		7			4		5	1		2		3

- Teams highlighted in purple go HAHAHA and AHAHA etc. This is useful for the cross league sharing scenario were one or maybe a few teams share grounds cross leagues.
- No team plays more than 2 consecutive home fixtures and no team therefore has more than 2 consecutive away fixtures
- Teams 1/6, 2/7, 3/8, 4/9, 5/10 are ‘magic pairs’ the first named team is away when the second named team in at home and vice versa. This is our key ground sharing attribute.
- The second half of fixtures repeats the order of the first half with the home and away teams transposed.
- The first 2 matches and last 2 matches have 1 home and 1 away match for all teams.

This is as good as it gets in terms of a fairness of competition for a league.

Extending this to multiple divisions

Having obtained a good template, the first thing to do for a multi divisions league is to use the same template in each division.

Note - if divisions are different sizes take a look at the section on this later in the document it becomes much harder.

The key thing is the ‘magic pairs’ mean the team in Slot 1 can always ground share with the team in Slot 4 and vice versa, 2 with 5, 2 with 6. This works equally as well cross the divisions because the template is the same as it does within one division.

Let’s look at a 3 division 6 team league and here is yet another grid. Don’t get this confused to the template in the previous section it is a grid of teams. Each team for that division needs to fit into a slot / row in the division. We are extending the concept of the ‘magic pairs’ so that if a team is allocated to slot 1 the complimentary team is still allocated to slot 4 but it just happened to be in a different column where column = division.

The completed solution looks little this

Div 1	Div 2	Div 3
Club A 1sts	Club A 2nds	Club J1
Club B 1sts	Club H	Club B 2nds
Club D1	Club C 1st	Club C 2nds
Club D2	Club E1	Club J2
Club F	Club E2	Club K
Club G	Club I	Club L

Club A, B, C have 1sts and 2nds and 1 ground each, Club D1 and D2 are different clubs but share a ground as do Clubs E1 and E2 and Club J1 and J2; the others clubs do not share grounds.

These teams which do not share grounds are very useful; they are our ‘Golden Teams’ and can be placed into any of the fixture patterns independently of any other team. In a sense they get what is left, but as all the patterns have the same degree of fairness in terms of meeting the basic requirements of the competition being placed last is not a disadvantage.

Here is the Blank Grid across the top is the divisions and down the side ‘slots’ for the teams. The symmetry of the solution however means we can use this across divisions.

	Div 1	Div 2	Div 3
Slot 1			
Slot 2			
Slot 3			
Slot 4			
Slot 5			
Slot 6			

Start by placing the Clubs with 1st and Seconds into the grid

	Div 1	Div 2	Div 3
Slot 1	Club A 1sts	Club C 1st	Club B 2nds
Slot 2			
Slot 3			
Slot 4	Club B 1sts	Club A 2nds	Club C 2nds
Slot 5			
Slot 6			

This could be done a number of ways but it can help to fill rows at a time if this is possible.

Slots 1 and 4 are magic pairs, Club A, Club B and Club C occupy rows 1 and 4 which guarantees there 1st XI are all at home when their 2nd XI are away.

What however would not work is if we first placed Club B 1sts in Slot 4 Div 1 meaning their 2nds are in Slot 1 in Div 3, then moved to Club C and tried to place their firsts into Slot 4 in Div 2. At that point Club C 2nds would need to be in Slot 1 in Div 3 but it is already occupied.

In the 18 team 6 teams a division it is not that hard by trial and error to resolve this move Team C somewhere else and carry on. As the number of teams and number of division increase and where all or most of the team share grounds this issue can get very time consuming to resolve.

Recently trying to do this in a 30 division by 10 teams league took over 132K operations of adding checking, removing teams and backtracking.

At this point it is worth considering the case of multiple teams across two leagues sharing grounds. Here it is possible even if fixtures are coordinated between leagues to fall into a problem which may not be immediately obvious. The league doing fixtures first places a team in slot 1 which determines in the pattern above the 2nd team of that club goes into slot 4. The first league then takes the next team and places them in slot 2 meaning the second team is in slot 5. This seems ok but the better approach would be for the first league to place the second team it allocates in slot 4 so that the team in the second league is in slot 1.

This is because the grid always fills up in pairs if we start with 2 empty spaces in the fourth row and three in the first row as the grid fills, we eventually end up with 1 slot left in the first row. We then do the same and also end up with one empty slot in the second row.

Those free slots in rows 1 and 2 can then only be occupied by teams with no sharing of grounds; the golden teams; and if there are not enough of these the fixtures will not work. A good tactic therefore is to keep the number of slots used in each of the magic pairs of rows the same.

Contrast to the higher league having used slots 1 and 4, then in the lower league the first and fourth rows have equal numbers of spaces and the allocations are more likely to work.

Then to continue to complete our grid, add the Clubs D1/D2, Clubs E1/E2and J1/J2 who share in the same division

	Div 1	Div 2	Div 3
Slot 1	Club A 1sts	Club C 1st	Club B 2nds
Slot 2	Club D1		Club J1
Slot 3		Club E1	
Slot 4	Club B 1sts	Club A 2nds	Club C 2nds
Slot 5	Club D2		Club J2
Slot 6		Club E2	

That is all the sharing dealt with in this simple league so fill in the grid with the rest of the teams

	Div 1	Div 2	Div 3
Slot 1	Club A 1sts	Club C 1st	Club B 2nds
Slot 2	Club D1	Club H	Club J1
Slot 3	Club F	Club E1	Club K
Slot 4	Club B 1sts	Club A 2nds	Club C 2nds
Slot 5	Club D2	Club I	Club J2
Slot 6	Club G	Club E2	Club L

Result when you apply the fixture template to each of these divisions then all ground shares are catered for and a good set of fixtures are produced.

Beyond the Basic requirements

The principle is that the more ‘special requirements’ there are, the less flexibility you get as these requirements eventually lead to conflicts which start to compromise something.

Combine this to the number of shared grounds you have and you can get a feel for the complexity. For instance, if all clubs share grounds in a division, then catering for 1 club’s need to play a particular sequence of games can be difficult.

The usual requirements like must play away on this date are sometimes not that hard to fulfil as this can be translated from club A must be away week 1 to club A must be in slot 3 ,4, or 5.

A share with another league can be translated to if I use slot 1 can you please use slot 4.

There are also some manipulations that could be done after a solution has been found to ‘force’ a team onto a particular pattern. You can do this using a combination of the following techniques:

- Swap (swapping two teams who are paired together with another pair who happen to be in the same divisions),
- Changing week 1 for week 2 and preserving the pattern by also week 6 with week 7 in our 6 team example.
- Exchange (exchanging two whole paired rows with another two paired rows e.g. in a 12-team pattern 2 and 8 with 5 and 11 (or 11 and 5))

We do however need to be aware of any other specific arrangements that you might have made when doing this and that things like no more than 2 home matches in a row might give.

Lastly it may be possible to take 1 fixture and reverse it if a team does not share a ground or move the date of that fixture.

General methodology

Fit the hard requirements in first. Say a team shares a ground with a team in another league and they do not use the same fixture template as you do. Most templates have a pattern which goes Home, Away, Home, Away etc. see if the 2 leagues have this and if so agree which league will use which sequence or alternatively find the best match, then add that team to your pattern first as there is little flexibility. In the example above use slot 3 or 8.

If you have a division where everyone shares, and another where only a few share, sort out the division where everyone shares first.

If a club needs a specific sequence of fixtures, take a look at the fixture template and see if the sequence exists and fit any such club in first.

If you have a league ladder where teams can be distributed anywhere in the ladder, look for a way to fill a row of slots, like the example above where Club A, Club B and Club C could fit into 1 slot.

If you do have fixture issues at the start or end of a season reverse 2 weeks, say week 2 and week 3 in the template. You may cause 3 home games or a slightly different order of playing but it may be a compromise which fixes the problem and it is easier than trying to change just one fixture. Note if you swap 2 weeks around in the template it affects **all your divisions**. Pick the friendlier club fixture secretaries and groundsman to give the 3 home matches on the trot to etc.

There are then some things which simply might not be possible. If a team tell you they must play all their home matches in August, it means 1 team in every division is going to do that and another team in every division is going to be away every fixture in August. If a team give you a specific list of Home fixtures it is unlikely without a lot of work a pattern will match this precisely.

Divisions with unequal number of teams

This can happen by design or happen if in lower divisions there are less teams.

The easy to resolve situation is divisions of 12 and then a division of 11 or 10 going to 9. This is because the 11 team division is a 12 team with a bye week and the 9 team league is using the 10 team template with a bye in it. Someone each week does not play in an odd team division and using the even template with a bye in it fixes the issue.

From a fixture generation point of view two 11 team divisions are easier to do than a 12 and a 10, however your clubs may not appreciate the blank dates. Having two 11 team divisions can often help, particularly at lower levels where one or more teams dropping out of a 10 team division can reduce the amount of cricket available to unacceptable levels.

Some leagues have 12 team divisions at the top and 10 team divisions at the bottom with there being 4 extra weeks in the 12 team divisions.

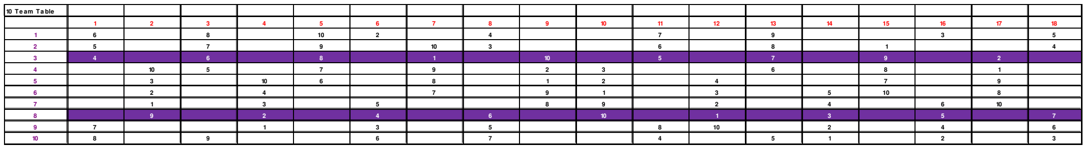
If there is very limited sharing between 12 and 10 team leagues then use the patterns of Home, Away, Home, Away to match between the 2 templates treating the problem as if two separate leagues.

If there is full sharing then something has to give.

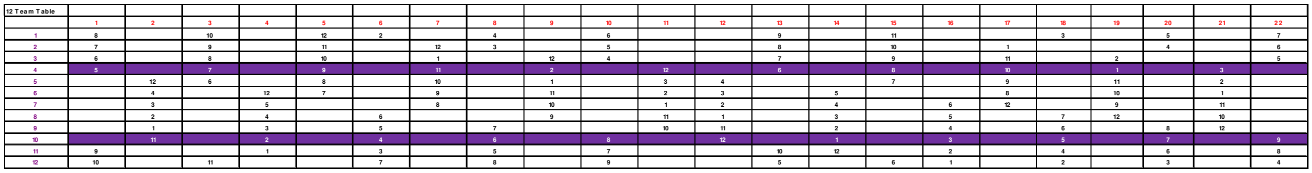
If we want a set of divisions with multiple shares in uneven sized divisions we need to be prepared to have to work hard to find the best set of templates to make this happen.

There is a solution below but it is not perfect.

Let’s have a quick look at the 10 and 12 team template.



10 Team Table																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	6		8		10	2		4			7		9			3		5
2	5		7		9		10	3			6		8		1			4
3	4		6		8		1		10		5		7		9		2	
4		10	5		7		9		2	3			6		8		1	
5		3		10	6		8		1	2		4			7		9	
6		2		4			7		9	1		3		5	10		8	
7		1		3		5			8	9		2		4		6	10	
8		9		2		4		6		10		1		3		5		7
9	7			1		3		5			8	10		2		4		6
10	8		9			6		7			4		5	1		2		3

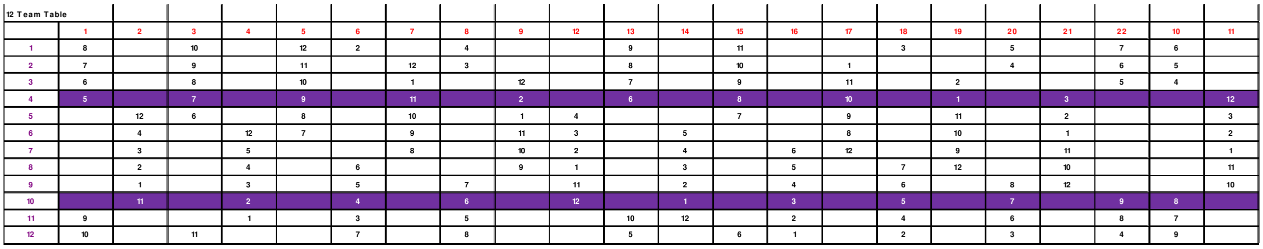


12 Team Table																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	8		10		12	2		4		6			9		11			3		5		7
2	7		9		11		12	3		5			8		10		1			4		6
3	6		8		10		1		12	4			7		9		11		2			5
4	5		7		9		11		2		12		6		8		10		1		3	
5		12	6		8		10		1		3	4			7		9		11		2	
6		4		12	7		9		11		2	3		5			8		10		1	
7		3		5			8		10		1	2		4		6	12		9		11	
8		2		4		6			9		11	1		3		5		7	12		10	
9		1		3		5		7			10	11		2		4		6		8	12	
10		11		2		4		6		8		12		1		3		5		7		9
11	9			1		3		5		7			10	12		2		4		6		8
12	10		11			7		8		9			5		6	1		2		3		4

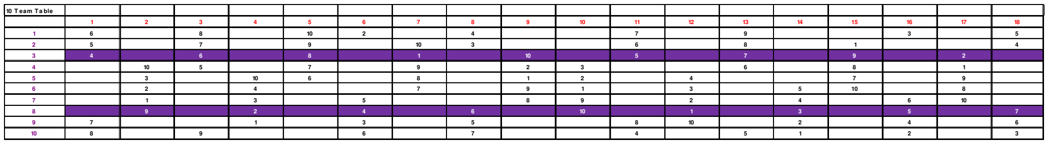
Don’t worry about the numbers just look at the patterns.

For the first 9 weeks all looks good Team 1 in the 10 team divisions is the same in terms of home and away patterns to the 12 team divisions, but it falls apart in week 11.

A simple fix is to take weeks 10 and 11 out of the 12 week template and place them at the end (or start).



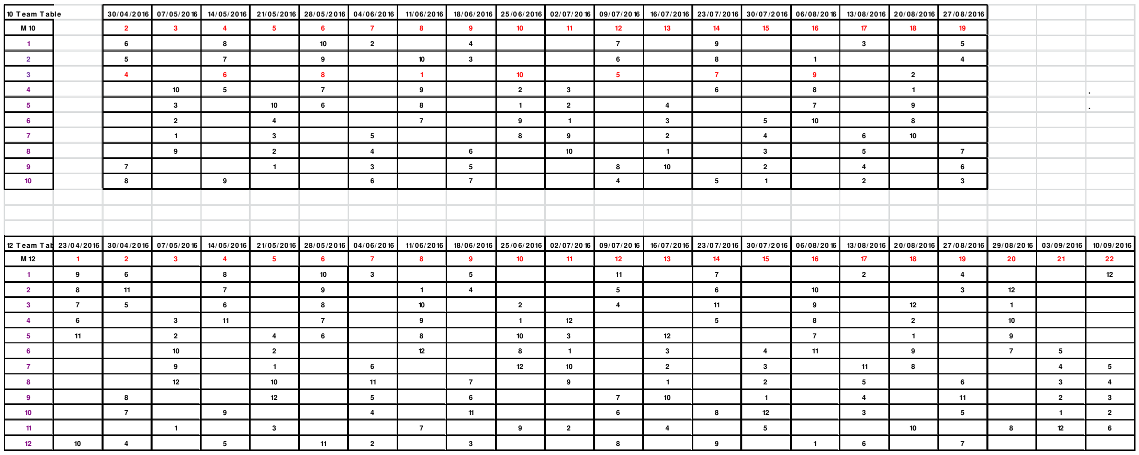
12 Team Table																						
	1	2	3	4	5	6	7	8	9	12	13	14	15	16	17	18	19	20	21	22	10	11
1	8		10		12	2		4			9		11			3		5		7	6	
2	7		9		11		12	3			8		10		1			4		6	5	
3	6		8		10		1		12		7		9		11		2			5	4	
4	5		7		9		11		2		6		8		10		1		3			12
5		12	6		8		10		1	4			7		9		11		2			3
6		4		12	7		9		11	3		5			8		10		1			2
7		3		5			8		10	2		4		6	12		9		11			1
8		2		4		6			9	1		3		5		7	12		10			11
9		1		3		5		7		11		2		4		6		8	12			10
10		11		2		4		6		12		1		3		5		7		9	8	
11	9			1		3		5			10	12		2		4		6		8	7	
12	10		11			7		8			5		6	1		2		3		4	9	



10 Team Table																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	6		8		10	2		4			7		9			3		5
2	5		7		9		10	3			6		8		1			4
3	4		6		8		1		10		5		7		9		2	
4		10	5		7		9		2	3			6		8		1	
5		3		10	6		8		1	2		4			7		9	
6		2		4			7		9	1		3		5	10		8	
7		1		3		5			8	9		2		4		6	10	
8		9		2		4		6		10		1		3		5		7
9	7			1		3		5			8	10		2		4		6
10	8		9			6		7			4		5	1		2		3

The patterns are now aligned again but the penalty is week 21 is reverse fixtures to 10 and week 22 to 11 so the symmetry of 1st half and second half of the 12 team league is destroyed.

An alternative solution is this one:



10 Team Table		30/04/2016	07/05/2016	14/05/2016	21/05/2016	28/05/2016	04/06/2016	11/06/2016	18/06/2016	25/06/2016	02/07/2016	09/07/2016	16/07/2016	23/07/2016	30/07/2016	06/08/2016	13/08/2016	20/08/2016	27/08/2016
M 10		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1		6		8		10	2		4			7		9			3		5
2		5		7		9		10	3			6		8		1			4
3		4		6		8		1		10		5		7		9		2	
4			10	5		7		9		2	3			6		8		1	
5			3		10	6		8		1	2		4			7		9	
6			2		4			7		9	1		3			10		8	
7			1		3		5			8	9		2		5	10		8	
8			9		2		4		6		10		1		3		5		7
9		7			1		3		5			8	10		2		4		6
10		8		9			6		7			4		5	1		2		3

12 Team Tab	23/04/2016	30/04/2016	07/05/2016	14/05/2016	21/05/2016	28/05/2016	04/06/2016	11/06/2016	18/06/2016	25/06/2016	02/07/2016	09/07/2016	16/07/2016	23/07/2016	30/07/2016	06/08/2016	13/08/2016	20/08/2016	27/08/2016	29/08/2016	03/09/2016	10/09/2016
M 12	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	9	6		8		10	3		5			11		7			2		4			12
2	8	11		7		9		1	4			5		6		10			3	12		
3	7	5		6		8		10		2		4		11		9		12		1		
4	6		3	11		7		9		1	12			5		8		2		10		
5	11		2		4	6		8		10	3		12			7		1		9		
6			10		2			12		8	1		3		4	11		9		7	5	
7			9		1		6			12	10		2		3		11	8			4	5
8			12		10		11		7		9		1		2		5		6		3	4
9		8			12		5		6			7	10		1		4		11		2	3
10		7		9			4		11			6		8	12		3		5		1	2
11			1		3			7		9	2		4		5			10		8	12	6
12	10	4		5		11	2		3			8		9		1	6		7			

Here during the season where 12 team division and 10 team division overlap, teams 1 thru 10 have the same pattern in both templates.

In the 12 team template Team 11 and Team 12 can share a ground.

In addition, although in the 12 team division the first and second halves of the season are not mirrors, the fixtures generated allowed a scheme of first 5 and last 6 are win lose and middle 11 are win lose draw to allow leagues to play half the season in one format and half in a second.

This second solution was obtained by spend time and effort moving the patterns around and transposing some fixtures to change home versus away until they lined up.

It was do for a specific set of league requirements.

A further solution which start from a position of will not compromise the 12 team league is this

M 12	1	2	3	4	5	6	8	9	10	11	12	13	14	15	16	17	18	19	20	21	23	24
1	9		12	2		4		11			8		10			3		5		6	7	
2	8		10		12	3		5			7		9		1			4		11	6	
3	7		9		1		12	4			6		8		10		2			5	11	
4	6		8		10		2		12		11		7		9		1		3		5	
5	11		7		9		1		3	4			6		8		10		2			12
6		5			7		9		1	2		4		11	12		8		10			3
7		4		11			8		10	1		3		5		6	12		9			2
8		3		5		6			9	10		2		4		11		7	12			1
9		2		4		11		7		12		1		3		5		6		8		10
10		1		3		5		6			9	12		2		4		11		7	8	
11		12	6		8		10		2	3		5			7		9		1			4
12	10			6		7		8			5		11	1		2		3		4	9	

M 10			3	4	5	6	8	9	10	11	12	13	14	15	16	17	18	19	20	21		
1			10	2		4		6			8		9			3		5		7		
2			9		10	3		5			7		8		1			4		6		
3			8		1		10	4			6		7		9		2			5		
4			7		9		2		10		5		6		8		1		3			.
5			6		8		1		3	4			10		7		9		2			.
6					7		9		2	3		4		5	10		8		1			
7				5			8		1	2		3		4		6	10		9			
8				4		6			9	1		2		3		5		7	10			
9				3		5		7		10		1		2		4		6		8		
10				6		7		8			9	5		1		2		3		4		

Here keeping the 12 team league ‘pure’ meant taking the middle weeks in the 10 team template and ends up with back to back fixtures in those divisions.

We are interested if anyone reading this document has a better solution to this ‘mixed sized division’ issue.

A tool to generate fixtures

For several years spreadsheet-based tools have been around which help take the templates and generate a set of fixtures, however matching the teams so that the teams sharing a ground has been a manual task or at best semi-automated.

On the web are several ‘free’ fixture generation tools but these appear restricted to a single divisions worth of fixtures.

The following description of a requirement but is illustrated with a working prototype coded solution albeit a solution not subjected to a rigorous set of testing with a few known bugs and therefore not yet viable for a general release.

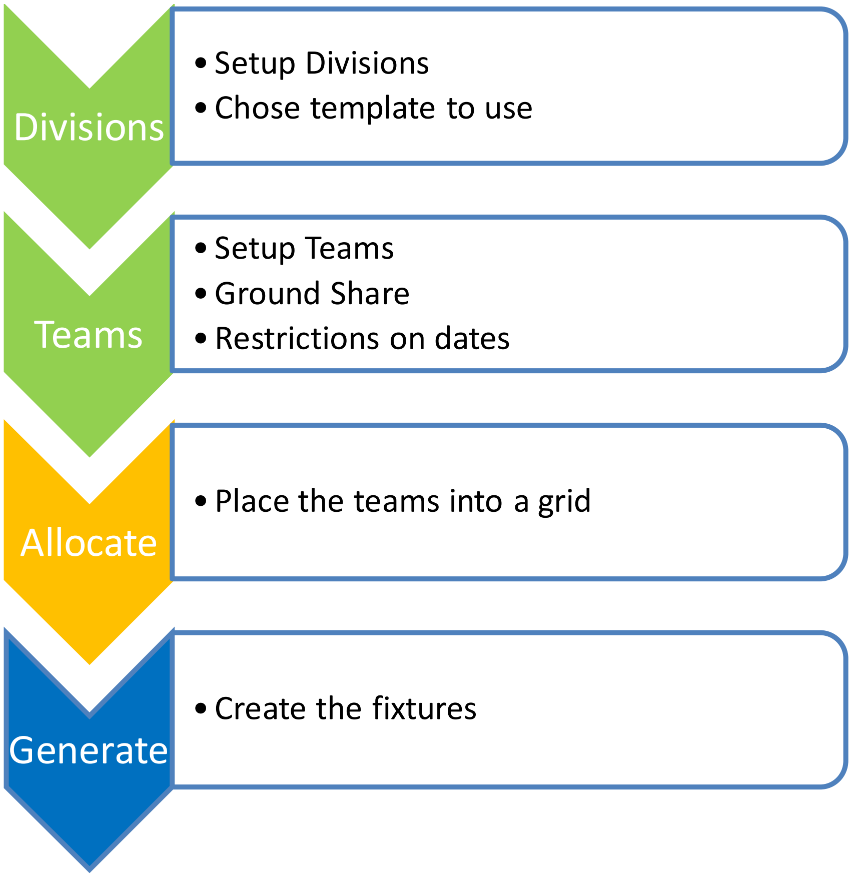
The requirement uses the prototype tool to illustrate how to generate fixtures using templates using either a manual approach with productivity aids or an automated tool.

There is also discussion of what this would like if integrated into the ECB Play-Cricket tool; that not being 100% necessary and as an alternative solution would be to take a feed in an agreed format from Play-Cricket (or any other competition management program) and import it, with fixtures exported in a similar agreed format at the end of the generation process.

Throughout this a Moscow approach of must do, should do, and could do is used. The prototype tool programmed does most of the key ‘must do tasks’ plus some additional ones but as it is not complete is missing a few ‘must do elements’.

General requirements - Overall Process

The main process flow is



- Setup Divisions
- Chose template to use

- Setup Teams
- Ground Share
- Restrictions on dates

- Place the teams into a grid

- Create the fixtures

Apart from this main process flow the following additional processes are needed.

1. A security process is needed with roles which would cover
    a. A fixture admin able to run the tool for 1 or more leagues
    b. A main admin able to administer the system as a whole including templates
2. An administrative function to allow new templates to be added for new styles of competitions. It may well be okay to allow for a single top level of administration of templates but it is desirable

General requirements –integration

The requirement is not for a full competition management tool but the solution should either closely or loosely integrate with such tool or tools. Options are

1) Integrate into a current competition management system making it tightly coupled to that system. For instance, incorporate it into Play-Cricket the ECB competition management tools
2) Run a derivative of the prototype tool stand alone with inputs and output either via CSV files or API calls to Play-Cricket or other systems

The above does not need to be an either / or it could be 'both'.

It should be noted that the current prototype tool takes Play-Cricket input in the form of exported system ids and ground shares then supplements that data before allocating the teams.

A hybrid approach could be possible where some features of Play-Cricket were enhanced, for instance the ground sharing code, an export was done to a stand-alone tool and then once fixtures were generated the Play-Cricket bulk fixture upload was used.

There are benefits from the all in one place approach of tightly coupling this feature into Play-Cricket but it may also be the case that doing so closes down possible usage to none Play-Cricket users. A possible reason not to do this could be that the tool becomes marketable to other sport or other countries for recreational sports fixtures.

General requirements UI and device

This is a specialist administration function and unlikely to be run on a mobile device. A modern UI aimed at a large screen user is required.

It should run on PC and Mac and require as little prerequisite software as possible. The current tools the ECB have are excel based which means any user needs a full excel license to use them and they do not run on Mac or Office 365 as macros are used.

It may well however be possible to run some of the functions on a high-end tablet device.

The current prototype tool is web based using MySQL, PHP and JavaScript and does significant client-side processing. This is not to dictate a solution set simply for information.

Specific requirements - Import leagues divisions and teams

To be able to generate fixtures the solution needs to have lists of divisions within the league, and the teams allocated to those divisions.

The prototype tool takes export of system ids and ground shares from Play-Cricket. It processes that input and creates database tables containing divisions and teams. It caters for multiple leagues and divisions by having a unique combination of league ID and Season.

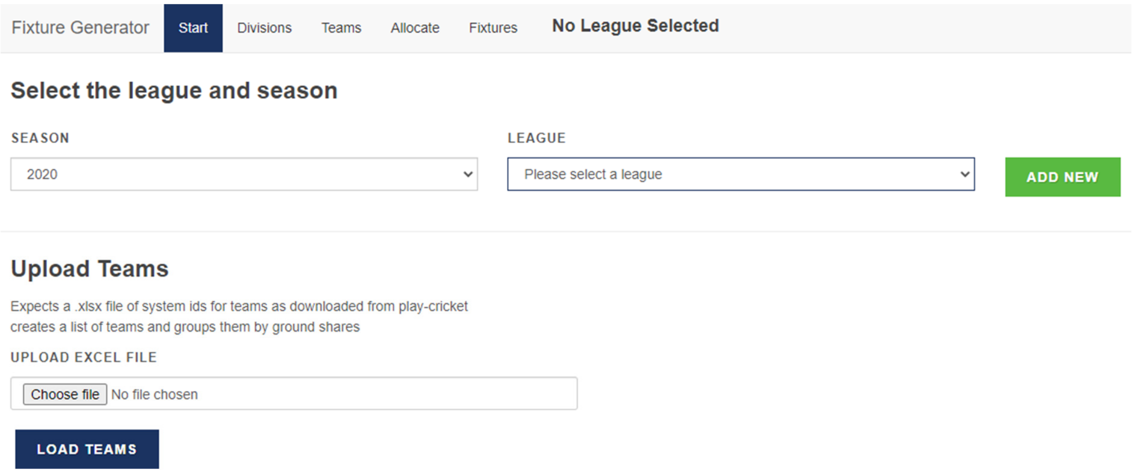
There is no restriction excepting the processing power of the client to the number of teams or divisions or the number of teams per division. Whilst the currently tested templates have up to 12 teams alternative templates for higher numbers and teams can be added.

Allowing for multiple versions of fixtures by league by year would also be desirable.

The fixture admin for the league should have permission to see all iterations relating to that league.

A main administrator should be able to see all iterations and all leagues

By way of example the prototype tool looks like this



No League Selected

Select the league and season

SEASON: 2020

LEAGUE: Please select a league

Upload Teams

Expects a .xlsx file of system ids for teams as downloaded from play-cricket
creates a list of teams and groups them by ground shares

UPLOAD EXCEL FILE: No file chosen

- It allows selection of Season and league
- It allows a new league to be added to the tool
- It uploads a file in the Play-Cricket system id export format into team and division tables

In this and other examples there has been a low level of design of the user interface which is functional but usability has not been extensively considered. The examples shown are not meant to be prescriptive in terms of the user interface.

Specific requirements - Setup the divisions templates and dates.

To be able to generate the fixtures some setup of the divisions is necessary. We need to select the templates, allocate them to the divisions and set up the dates and times that the fixtures will relate to.

Templates

The fixtures will be generated from templates.

Therefore, a prerequisite is a template for that league exists. Many leagues are similar formats for instance

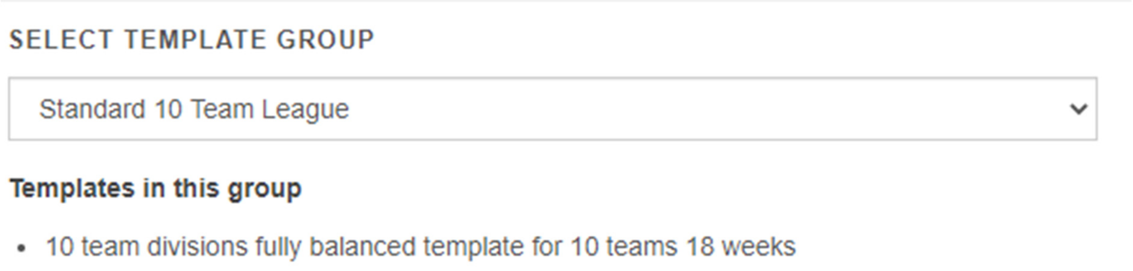
- Multiple 10 team divisions
- Multiple 12 team divisions

The major attribute of a fixture template is the number of teams. A 12 team template for instance is no good for a 10 team league.

There are a significant number of cases where there are different size divisions in a league and noting the restrictions discussed previously this need to be catered for.

To do this it is suggested that there is a concept of a template group where each template in the group matches to the other templates in the same group such that any template which matches the division size in that group can be chosen.

In the simple case



SELECT TEMPLATE GROUP

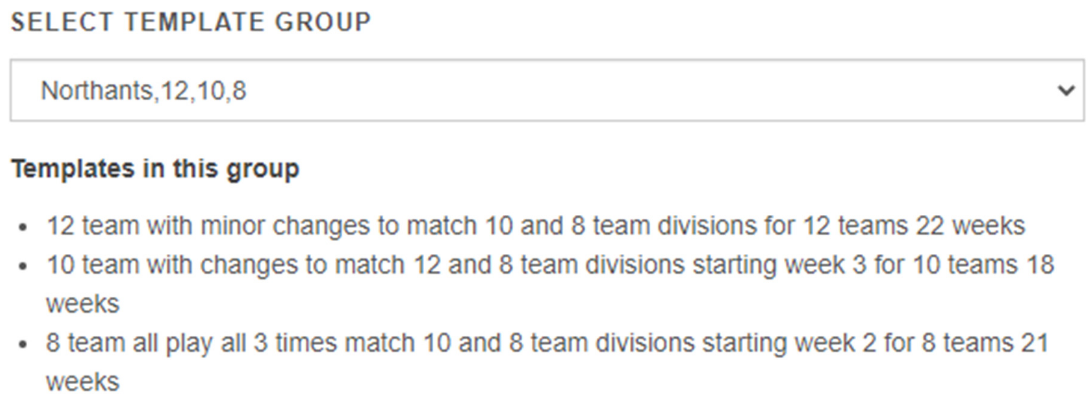
Standard 10 Team League

Templates in this group

- 10 team divisions fully balanced template for 10 teams 18 weeks

There might be only a single template in the group as above which by default would apply to all divisions.

In more complex cases a league might have several templates relating to differing division formats



SELECT TEMPLATE GROUP

Northants,12,10,8

Templates in this group

- 12 team with minor changes to match 10 and 8 team divisions for 12 teams 22 weeks
- 10 team with changes to match 12 and 8 team divisions starting week 3 for 10 teams 18 weeks
- 8 team all play all 3 times match 10 and 8 team divisions starting week 2 for 8 teams 21 weeks

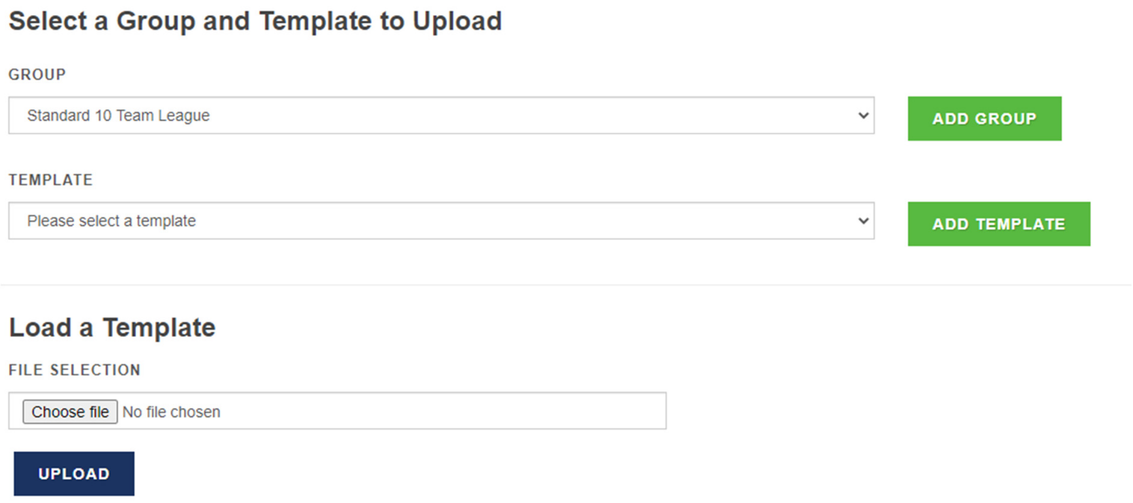
Based on the division structure it should be possible to identify which template group or groups will work with any league.

It should also be possible to assign the 'best fit' template from a group to a division based on the template having the correct number of teams and where more than 1 template fits the criteria allocating the first found as default.

Loading new templates and groups.

Whilst this is an admin function which would not be required often, the ability to setup new templates and template groups and load these at a top-level administration level is desirable, with the alternative being developers would need to control this function.

A facility similar to that in the prototype tool is needed



Select a Group and Template to Upload

GROUP

Standard 10 Team League

TEMPLATE

Please select a template

Load a Template

FILE SELECTION

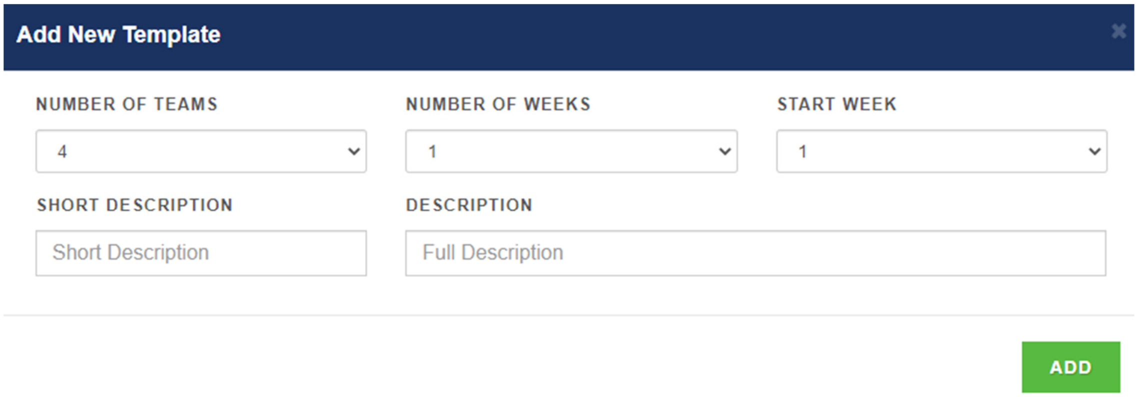
Choose file No file chosen

A template should have the attributes of

- Number of teams
- Number of Rounds (or Weeks)
- And the start week

This then allows templates of different sizes to be aligned. For instance, a 12 team template of 22 rounds starting at week 1 can be aligned with a 10 team template of 18 weeks starting at week 3. This would mean weeks 3 thru 20 were common between the templates.

A sample add template screen could look like



Add New Template

NUMBER OF TEAMS: 4

NUMBER OF WEEKS: 1

START WEEK: 1

SHORT DESCRIPTION: Short Description

DESCRIPTION: Full Description

The templates should be as generic as possible and therefore should be defined as ‘rounds’ or ‘week numbers’ rather than specific dates.

A suggested format for the upload files for a template is

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	2		6		10	3		7			4		8			5		9
3	9		4		8		10	5			2		6		1			7
5	7		2		6		1		10		9		4		8		3	
7		10	9		4		8		3	5			2		6		1	
9		5			2		6		1	3		7	10		4		8	
2		3		7			4		8	1		5		9	10		6	
4		1		5		9			6	8		3		7		2	10	
6		8		3		7		2		10		1		5		9		4
8	4			1		5		9			6	10		3		7		2
10	6		8	10		2		4			7			1		3		5

Note this is a similar template to that presented in the discussion on how templates work. For the sample tool to make the number of teams generic the templates are laid out such that odd / even pair is a 'magic pair' such that 1 is at home 2 is away etc. This appeared an easier way to design a tool for a variable set of divisions. In a 6 team division 1/2, 3/4, 5/6 are the magic pairs, then in a 8 team division add 7/8 as the next pair and so on.

Optional features in a template uploader would be

- A validation that the template met the basic requirement criteria for a template and was complete
- The ability to upload the template name and description with the template

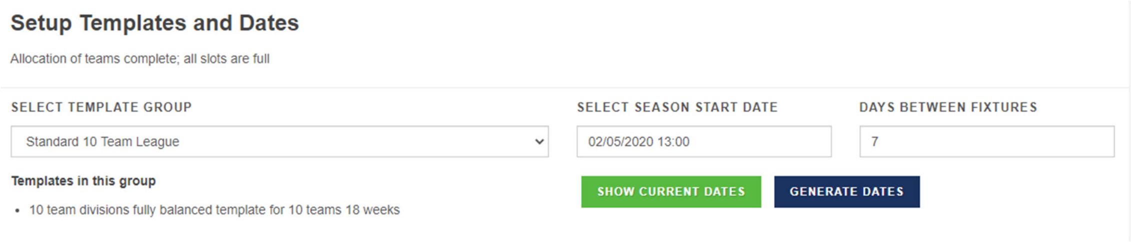
In addition, the prototype tool only allows a template to be used in a single template group. This restriction should be removed by implementing a cross reference allowing a many to many relationships between templates and template groups. This would prevent the need to upload the template multiple times once for each group it was part of.

Specific requirements - Dates and times

The fixtures generated will need to be associated with dates and times. To keep the templates generic, they should relate to rounds or weeks.

Therefore, a way for the fixture admin to assign dates / times to the template rounds is needed.

This could work something like



Setup Templates and Dates

Allocation of teams complete; all slots are full

SELECT TEMPLATE GROUP: Standard 10 Team League

SELECT SEASON START DATE: 02/05/2020 13:00

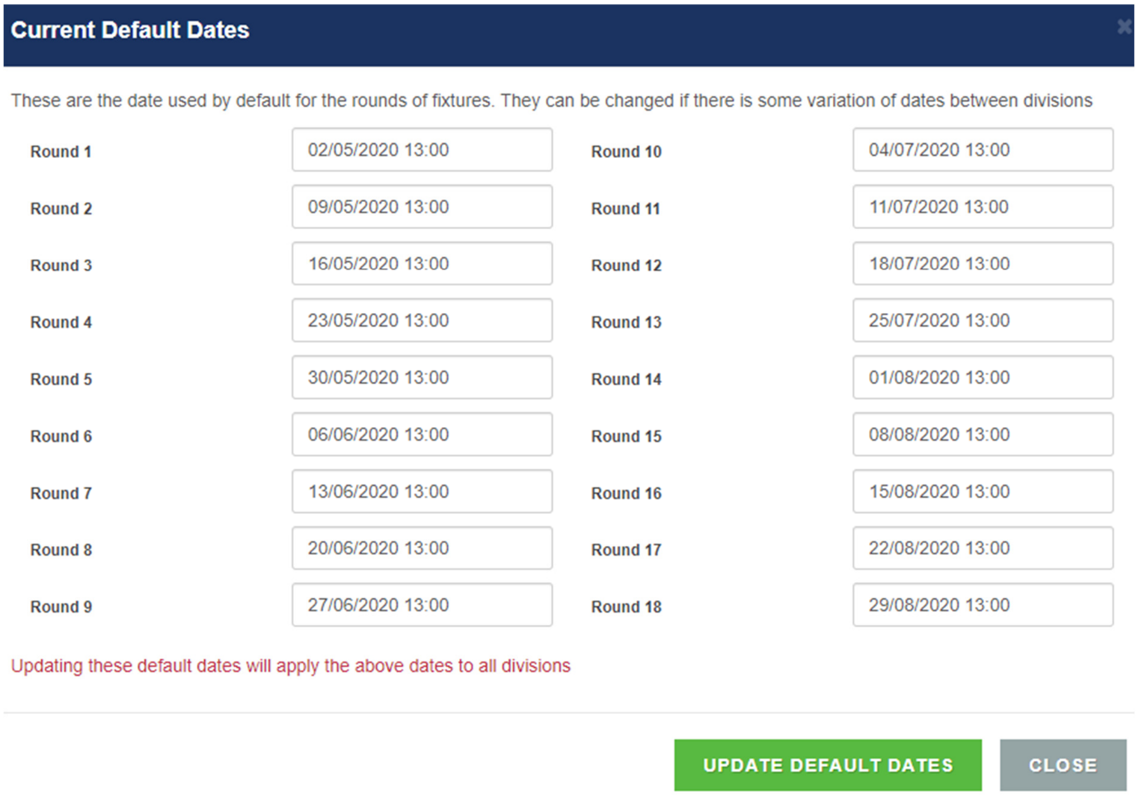
DAYS BETWEEN FIXTURES: 7

Templates in this group

- 10 team divisions fully balanced template for 10 teams 18 weeks

In many cases a season will run on a same set day each week so specifying the start date and gap between fixtures defaulting at 7 days will allow a quick allocation of date/times to rounds.

In the prototype tool if the start date and days between fields are completed a list of dates can be generated. Press the generate dates button.



Current Default Dates

These are the date used by default for the rounds of fixtures. They can be changed if there is some variation of dates between divisions

Round 1	02/05/2020 13:00	Round 10	04/07/2020 13:00
Round 2	09/05/2020 13:00	Round 11	11/07/2020 13:00
Round 3	16/05/2020 13:00	Round 12	18/07/2020 13:00
Round 4	23/05/2020 13:00	Round 13	25/07/2020 13:00
Round 5	30/05/2020 13:00	Round 14	01/08/2020 13:00
Round 6	06/06/2020 13:00	Round 15	08/08/2020 13:00
Round 7	13/06/2020 13:00	Round 16	15/08/2020 13:00
Round 8	20/06/2020 13:00	Round 17	22/08/2020 13:00
Round 9	27/06/2020 13:00	Round 18	29/08/2020 13:00

Updating these default dates will apply the above dates to all divisions

This kind of approach then allows the fixtures administrator to alter the date of a particular round or rounds (say round 18 was move to a bank holiday Monday) or the league wanted to reorder the rounds so round 2 was played first before round 1.

A common requirement also might be rounds before or after a certain date have matches starting earlier than the normal start. The objective of this feature is to try to minimise the amount of changes that the fixture administrator has to enter so if there are ways to minimise keying of data these should be considered.

In the example the update default dates change the whole league, confirming the admin wants to make this change is desirable.

There is also a feature described later to change dates and times for a single division.

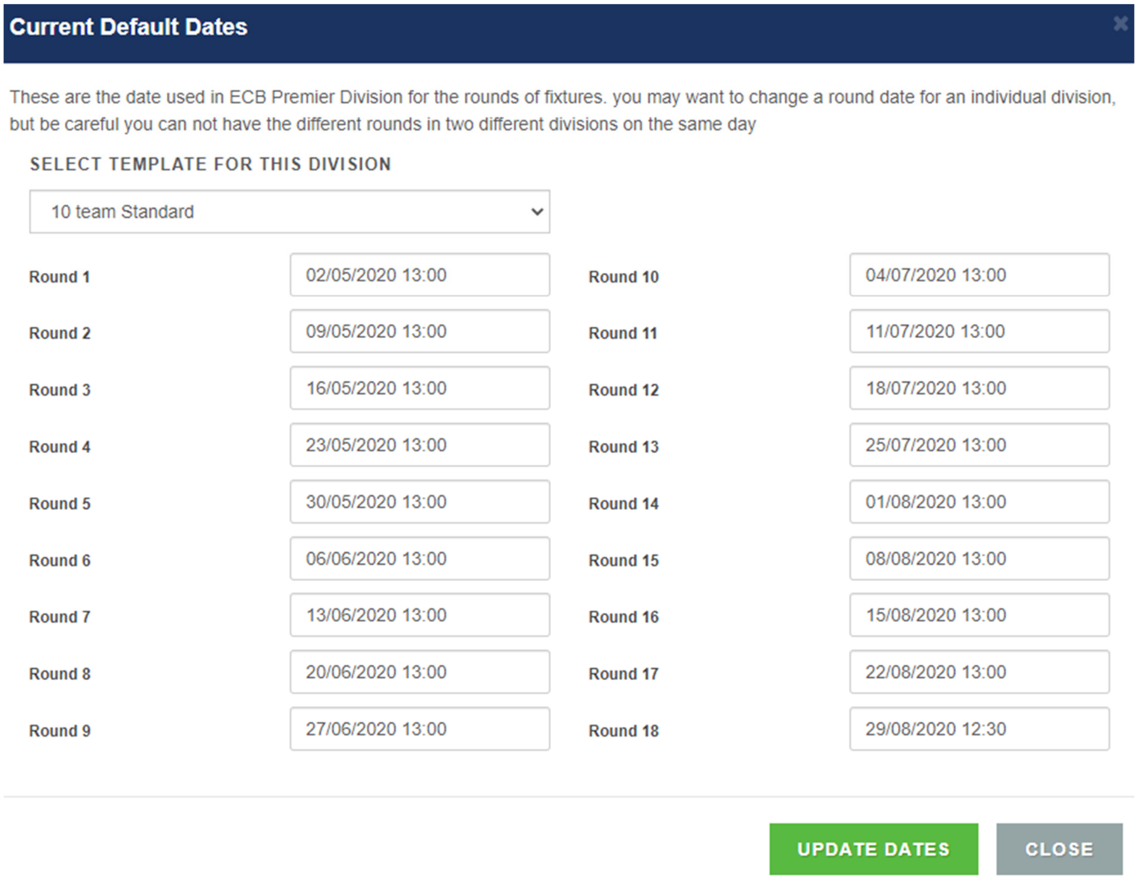
Specific requirements - Division setup

In the prototype tool the division management is separate to that in the competition management tool.

Even if the generation of fixtures sat outside Play-Cricket consideration could be given to updating the completion management of divisions to add the fixture template selection and dates selection into competition management.

Whilst for some leagues all divisions will start at the same time, play all the same dates There may be some differences by division in some leagues. This must be allowed for.

As an example, by division it should be possible to edit the template being used (selecting an alternate from the same group) and alter the match dates and times



Current Default Dates

These are the date used in ECB Premier Division for the rounds of fixtures. you may want to change a round date for an individual division, but be careful you can not have the different rounds in two different divisions on the same day

SELECT TEMPLATE FOR THIS DIVISION

10 team Standard

Round 1	02/05/2020 13:00	Round 10	04/07/2020 13:00
Round 2	09/05/2020 13:00	Round 11	11/07/2020 13:00
Round 3	16/05/2020 13:00	Round 12	18/07/2020 13:00
Round 4	23/05/2020 13:00	Round 13	25/07/2020 13:00
Round 5	30/05/2020 13:00	Round 14	01/08/2020 13:00
Round 6	06/06/2020 13:00	Round 15	08/08/2020 13:00
Round 7	13/06/2020 13:00	Round 16	15/08/2020 13:00
Round 8	20/06/2020 13:00	Round 17	22/08/2020 13:00
Round 9	27/06/2020 13:00	Round 18	29/08/2020 12:30

If the date of a round is altered there should be some rules applied to ensure that the integrity of the fixtures across the league is met. For instance if the default in to play round 1 on 1st May and the fixtures admin in one of the divisions determines round 2 will be 1st May this will not work whereas if for 1 division round 1 was moved to 2nd May which is not used by any other round then that will work.

The rule should be that within the league different rounds in different division should not overlap and appear at the same date and time. Note in some circumstance 2 rounds might occur to the same day.

A real example of the need for this is where a county league feeds a premier league via a play off system. The county league ends one week earlier than the premier league by moving the last round of fixtures forward to August Bank Holiday Monday.

Specific requirements - Team setup

The teams in the league in the prototype tool are loaded from the Play-Cricket export of the system ids.

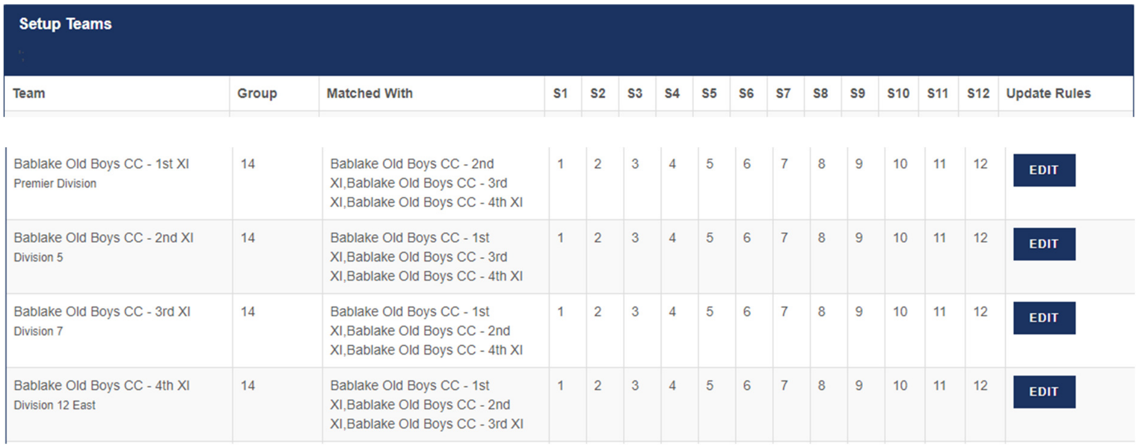
Even if the generation tool is left to be stand alone there is some benefit in amending this export and adding more information to play cricket.

The prototype tool takes the play cricket system ids and ground shares and introduces the concept of groups of teams.

Basic shared grounds allow for Team A shares with Team B. The difference in the group approach is that if say Team A must have the opposite fixtures as Team B (a ground share) if also allows Team C to be added to the group with the same fixtures as Team B.

Real life examples of this is Bablake CC or Kidderminster CC were the configuration of the venue means 2 ground are adjacent. The main ground is used by 1st XI and 2nd XI and when the 1st XI is at home then the 4th XI should also be a home, meaning the 3rd XI should be away such that the club can maximise the playing area for their top level matches.

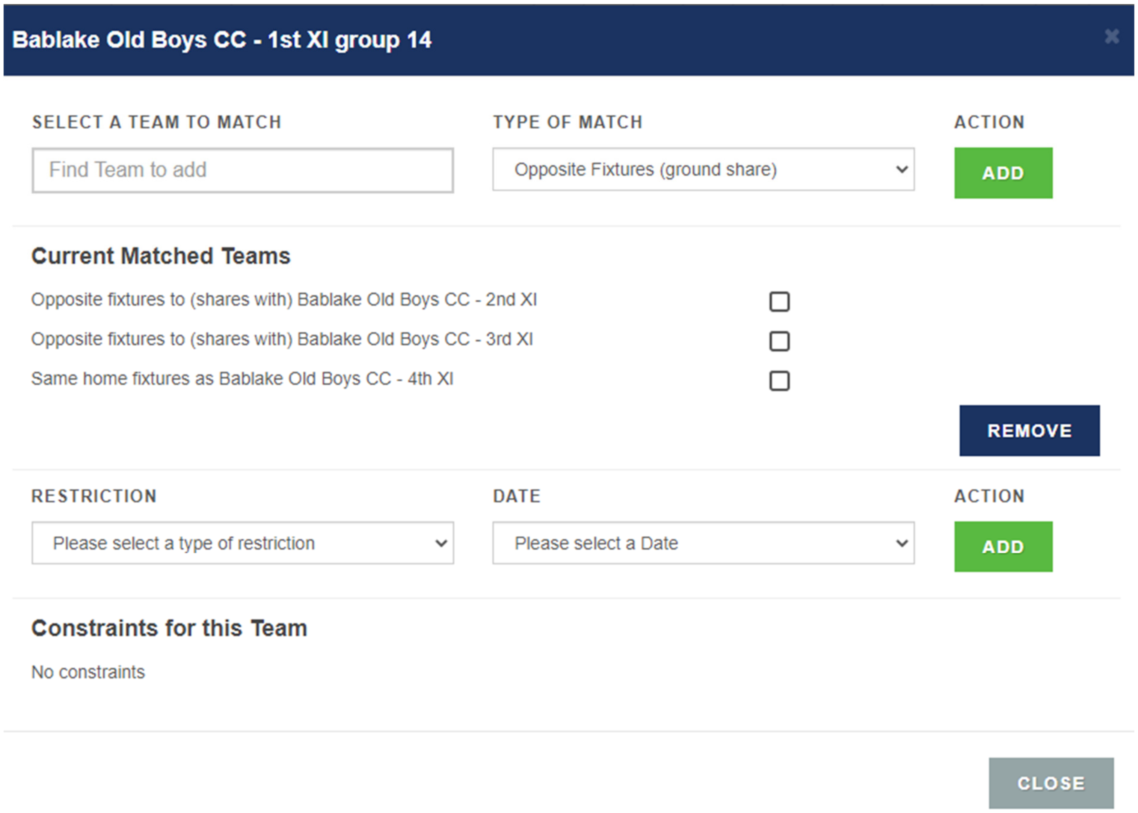
Within the prototype tool this looks like



Setup Teams

Team	Group	Matched With	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	Update Rules
Bablake Old Boys CC - 1st XI Premier Division	14	Bablake Old Boys CC - 2nd XI,Bablake Old Boys CC - 3rd XI,Bablake Old Boys CC - 4th XI	1	2	3	4	5	6	7	8	9	10	11	12	EDIT
Bablake Old Boys CC - 2nd XI Division 5	14	Bablake Old Boys CC - 1st XI,Bablake Old Boys CC - 3rd XI,Bablake Old Boys CC - 4th XI	1	2	3	4	5	6	7	8	9	10	11	12	EDIT
Bablake Old Boys CC - 3rd XI Division 7	14	Bablake Old Boys CC - 1st XI,Bablake Old Boys CC - 2nd XI,Bablake Old Boys CC - 4th XI	1	2	3	4	5	6	7	8	9	10	11	12	EDIT
Bablake Old Boys CC - 4th XI Division 12 East	14	Bablake Old Boys CC - 1st XI,Bablake Old Boys CC - 2nd XI,Bablake Old Boys CC - 3rd XI	1	2	3	4	5	6	7	8	9	10	11	12	EDIT

All 4 Bablake teams are in the same group. This is maintained from a screen used to edit the restrictions.



Bablake Old Boys CC - 1st XI group 14

SELECT A TEAM TO MATCH	TYPE OF MATCH	ACTION
Find Team to add	Opposite Fixtures (ground share)	ADD

Current Matched Teams

- Opposite fixtures to (shares with) Bablake Old Boys CC - 2nd XI □
- Opposite fixtures to (shares with) Bablake Old Boys CC - 3rd XI □
- Same home fixtures as Bablake Old Boys CC - 4th XI □

RESTRICTION	DATE	ACTION
Please select a type of restriction	Please select a Date	ADD

Constraints for this Team

No constraints

See the top half of this screen

Teams can be added or removed from a group.

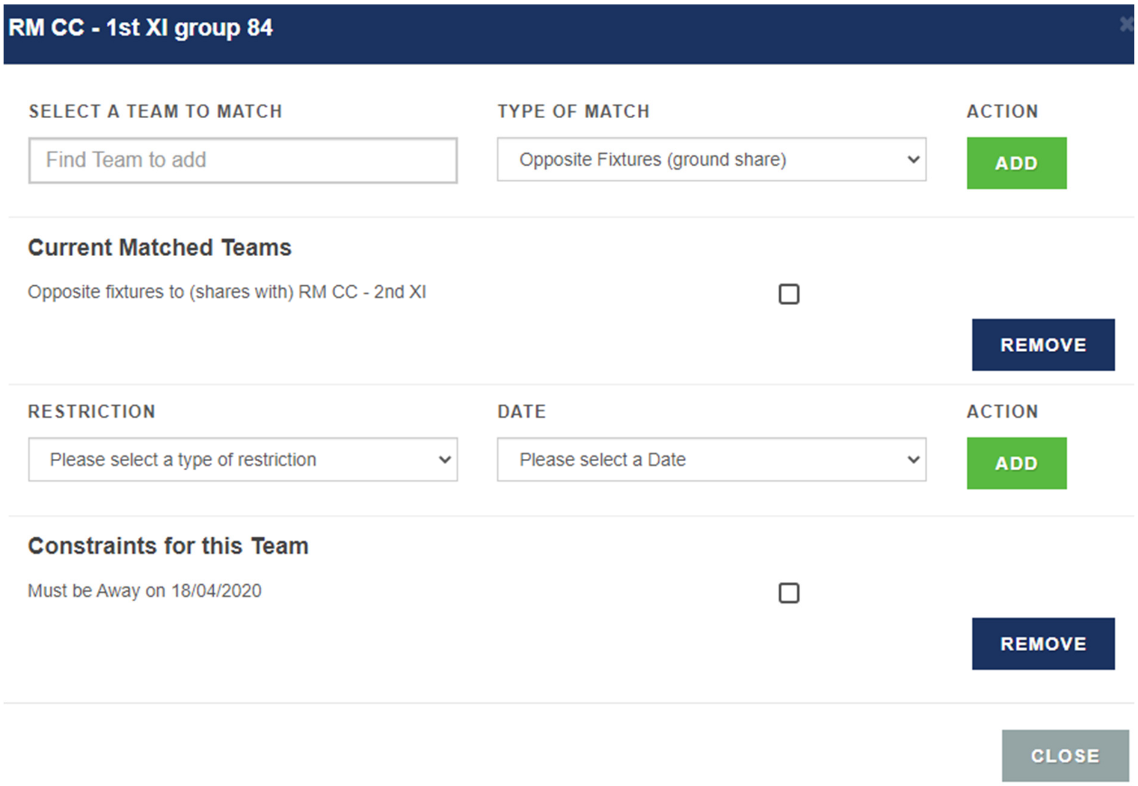
Then the bottom part of the screen has other common restrictions placed on leagues by clubs are

- Can not play at home on this date
- Must match to a club in another league

Play-Cricket would need enhancement as it allows 1 team to be linked only with 1 other, and does not allow for any on the 'no available' constraints.

Selecting must be at home or away on a specific date restricts the possible placement of the team in the fixture generation.

For instance



RM CC - 1st XI group 84

SELECT A TEAM TO MATCH: Find Team to add

TYPE OF MATCH: Opposite Fixtures (ground share)

Current Matched Teams

Opposite fixtures to (shares with) RM CC - 2nd XI □

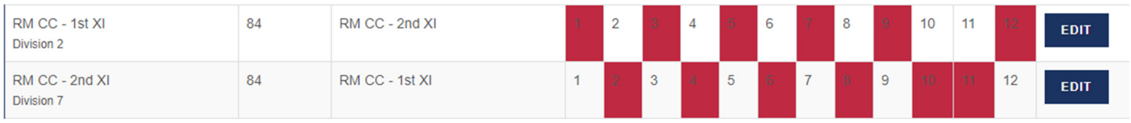
RESTRICTION: Please select a type of restriction

DATE: Please select a Date

Constraints for this Team

Must be Away on 18/04/2020 □

RMCC must be away on 18th April results in half the available slots being removed



RM CC - 1st XI Division 2	84	RM CC - 2nd XI	🟥 1	2	🟥 3	4	🟥 5	6	🟥 7	8	🟥 9	10	11	🟥 12	EDIT
RM CC - 2nd XI Division 7	84	RM CC - 1st XI	1	🟥 2	3	🟥 4	5	🟥 6	7	🟥 8	9	🟥 10	🟥 11	12	EDIT

It is possible if a club asks the league for too much flexibility then the number of available combinations goes to zero.

This visually here would result in all the slots showing red.

Some things are impossible, or would need significant changes affecting multiple clubs by changing the template. For instance, if RMCC needed to play away on both 18th and 25th April this would not work because the basic principle of the template was the each team plays once at home and once away in the first 2 matches.

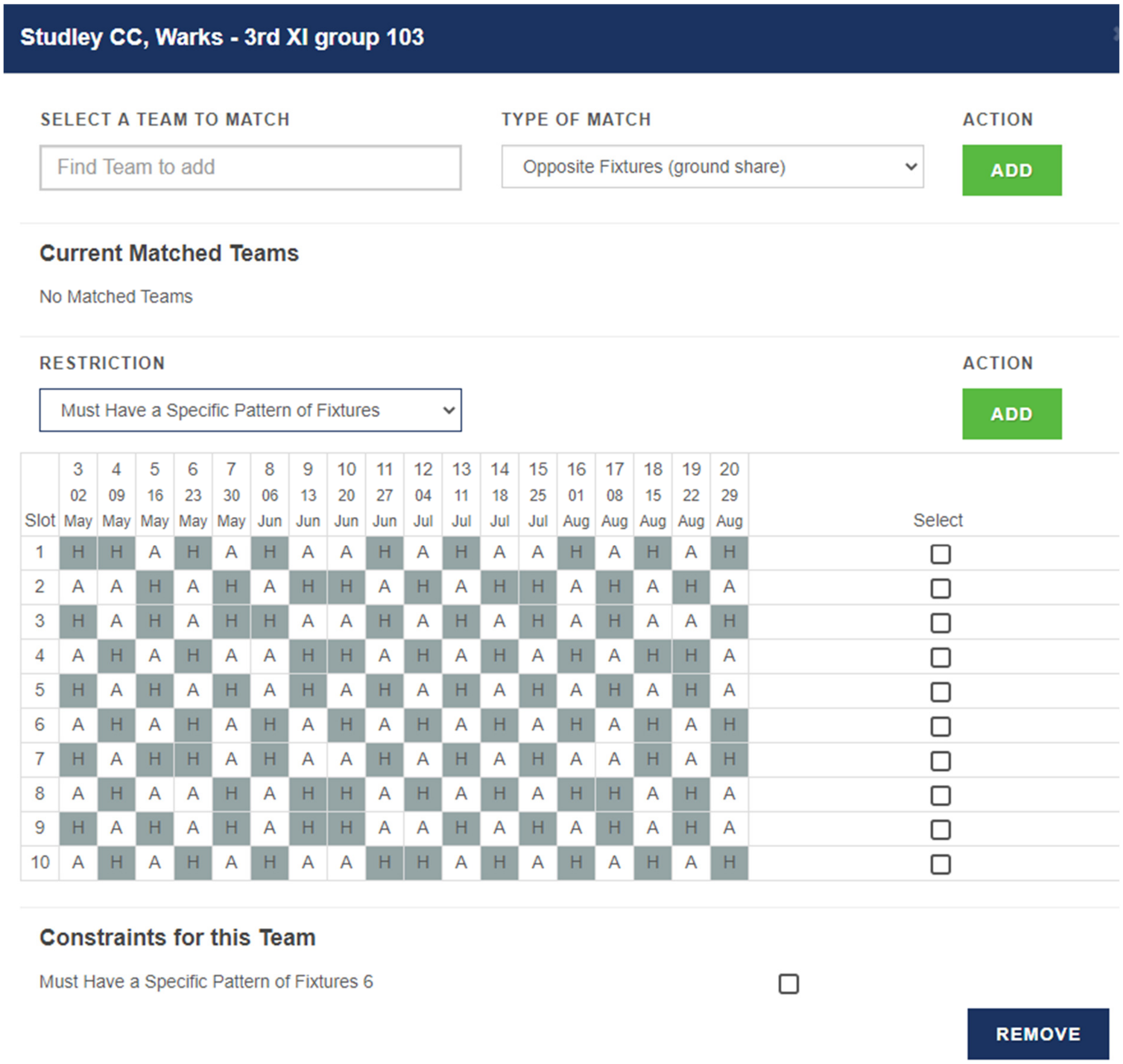
Whilst this could be resolved within the fixture generate by swapping the dates for week 2 and week 3. The league needs then to determine if to allow this at the expense of a small compromise to the perceived fairness of the competition.

The prototype tool and requirements from this document would allow a league to easily reverse week 2 and week 3 so if they wish they can resolve some of these issues.

The prototype tool illustrates when a set of constraints mean that no positions in the template will resolve the problem by turning all available slots for a team red.

An alternative approach alerting the fixture admin when a combination of constraints makes completion of the fixtures impossible could be considered.

If a team shares with another team from another league that team may need to occupy a fixed place in the template for that division. This should be allowed for in the generation tool.



Studley CC, Warks - 3rd XI group 103

SELECT A TEAM TO MATCH: Find Team to add

TYPE OF MATCH: Opposite Fixtures (ground share)

ACTION: ADD

Current Matched Teams

No Matched Teams

RESTRICTION: Must Have a Specific Pattern of Fixtures

ACTION: ADD

	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
	02	09	16	23	30	06	13	20	27	04	11	18	25	01	08	15	22	29	
Slot	May	May	May	May	May	Jun	Jun	Jun	Jun	Jul	Jul	Jul	Jul	Aug	Aug	Aug	Aug	Aug	Select
1	H	H	A	H	A	H	A	A	H	A	H	A	A	H	A	H	A	H	□
2	A	A	H	A	H	A	H	H	A	H	A	H	H	A	H	A	H	A	□
3	H	A	H	A	H	H	A	A	H	A	H	A	H	A	H	A	A	H	□
4	A	H	A	H	A	A	H	H	A	H	A	H	A	H	A	H	H	A	□
5	H	A	H	A	H	A	H	A	H	A	H	A	H	A	H	A	H	A	□
6	A	H	A	H	A	H	A	H	A	H	A	H	A	H	A	H	A	H	□
7	H	A	H	H	A	H	A	A	H	A	H	A	H	A	A	H	A	H	□
8	A	H	A	A	H	A	H	H	A	H	A	H	A	H	H	A	H	A	□
9	H	A	H	A	H	A	H	H	A	A	H	A	H	A	H	A	H	A	□
10	A	H	A	H	A	H	A	A	H	H	A	H	A	H	A	H	A	H	□

Constraints for this Team

Must Have a Specific Pattern of Fixtures 6 □

REMOVE

Here Studley share with a team in another league. The 2 fixture admins have agreed that Studley will be home and away alternate weeks and stat with an away fixture. The pattern for this is Slot 6.

The fixture admin can see the pattern for each slot and pick one or more that match.

This facility can also be used to fixture interconnected leagues where say the first XI plays in one league and the 2nd XI in another.

Typically but not necessarily the 1st XI fixtures would be sorted out first and this would determine the positioning of the 2nd XI, noting fixture admins would need to agree to use the same groups of templates.

In some leagues the number of constraints will be low whilst in other virtually every team will share and some will need very specific dates.

Allocating Teams to Slots

There are tools already available some published and some home written which help fixture admins add teams to the slots.

Within Play-Cricket there is an external spreadsheet system written in 2016 which aids this.

These tools help the fixture admin by preventing clashes. The Play-Cricket tool is currently limited to dealing with ground sharing. If a 1st team is placed into a slot in the top row, then the tool places the 2nd team; sharing a ground into the complimentary slot.

The prototype tool has the same approach of a grid with divisions across the top and rows. It is arranged as a set on odd and even pairs. If a team is placed in slot 1 in the first division then automatically the tool places the team ground sharing into slot 2 in its division.

This is then extended from the current spreadsheet approach in the current way

- Other teams in the same group are also placed
- The constraints on dates previously setup are converted to slots that are unavailable and also tracked

The objective is the tool tracks all the constraints and shares and no additional ‘pen and paper’ process is necessary.

An enhancement would also be to try to cater for the ground not available constraint by matching the team with this constraint to be at home against a bye or golden team.

This is possible in the prototype tool but only by forcing both the team concerned and the bye/golden team into specific slots in the grid.

General characteristics

The prototype tool runs client side on a web browser.

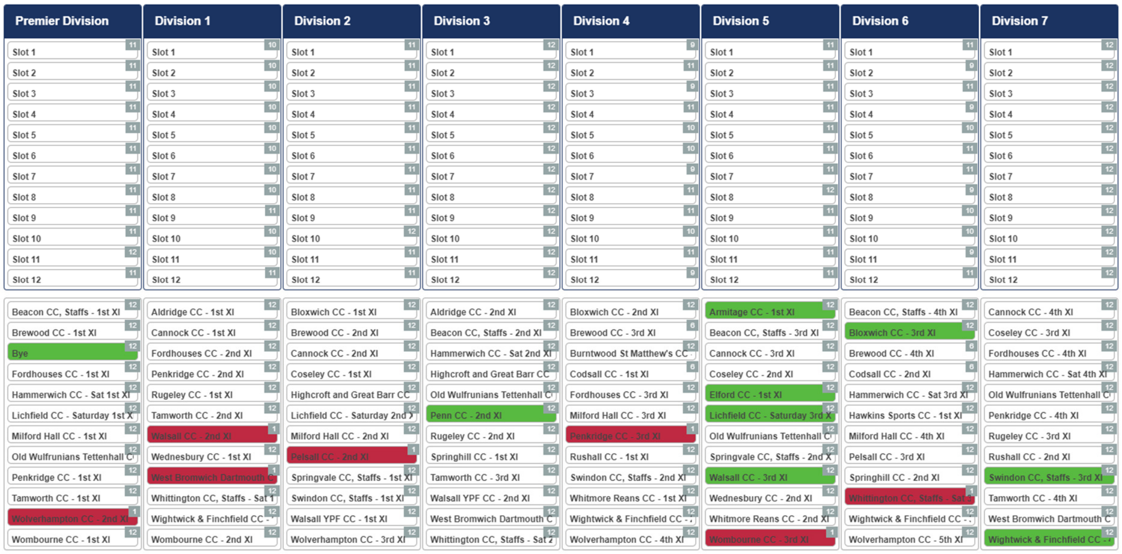
The choice of running client side is arbitrary and there are good arguments for both local and client-side processing.

The user interface is not prescriptive. The examples use an interface which is a grid. The wider the screen the easier it is to use the grid. Other approaches are equally valid.

The user in the prototype tool can drag and drop from a list at the bottom of the screen into the allocations grid at the top of the screen, this works for around 12 teams per division but again is not the only valid method.

The key objective is to place all the teams into positions in this case the grid at the top half of the screen in a way that for each team all the constraints are met.

Productivity aids



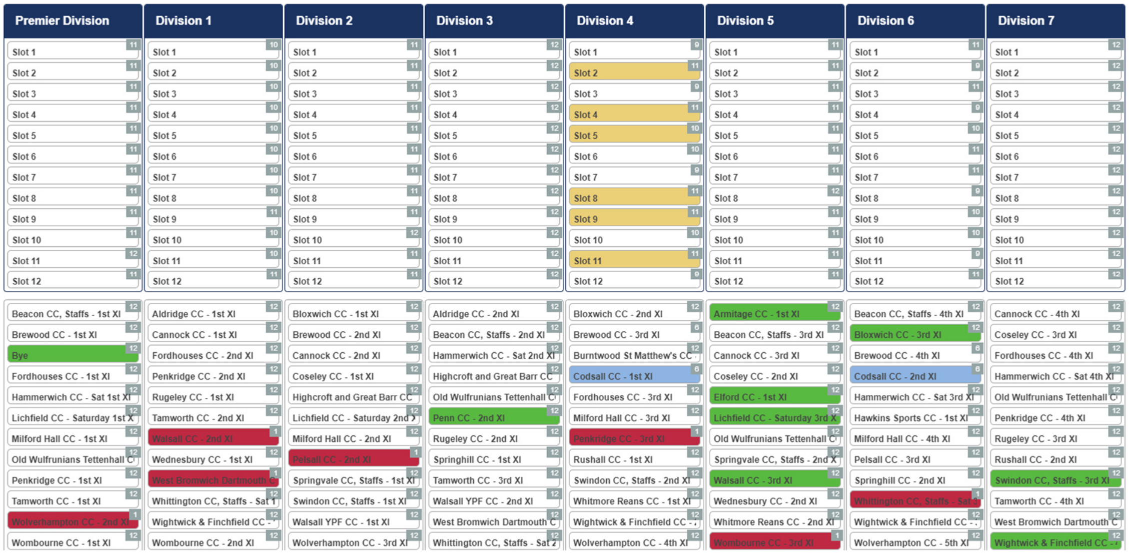
Premier Division	Division 1	Division 2	Division 3	Division 4	Division 5	Division 6	Division 7
Slot 1 11	Slot 1 10	Slot 1 11	Slot 1 12	Slot 1 9	Slot 1 11	Slot 1 11	Slot 1 12
Slot 2 11	Slot 2 10	Slot 2 11	Slot 2 12	Slot 2 11	Slot 2 11	Slot 2 9	Slot 2 12
Slot 3 11	Slot 3 10	Slot 3 11	Slot 3 12	Slot 3 9	Slot 3 11	Slot 3 11	Slot 3 12
Slot 4 11	Slot 4 10	Slot 4 11	Slot 4 12	Slot 4 11	Slot 4 11	Slot 4 9	Slot 4 12
Slot 5 11	Slot 5 10	Slot 5 11	Slot 5 12	Slot 5 10	Slot 5 11	Slot 5 10	Slot 5 12
Slot 6 11	Slot 6 10	Slot 6 11	Slot 6 12	Slot 6 10	Slot 6 11	Slot 6 11	Slot 6 12
Slot 7 11	Slot 7 10	Slot 7 11	Slot 7 12	Slot 7 9	Slot 7 11	Slot 7 11	Slot 7 12
Slot 8 11	Slot 8 10	Slot 8 11	Slot 8 12	Slot 8 11	Slot 8 12	Slot 8 9	Slot 8 12
Slot 9 11	Slot 9 11	Slot 9 11	Slot 9 12	Slot 9 11	Slot 9 11	Slot 9 10	Slot 9 12
Slot 10 11	Slot 10 10	Slot 10 12	Slot 10 12	Slot 10 10	Slot 10 11	Slot 10 10	Slot 10 12
Slot 11 12	Slot 11 10	Slot 11 11	Slot 11 12	Slot 11 11	Slot 11 11	Slot 11 9	Slot 11 12
Slot 12 11	Slot 12 11	Slot 12 11	Slot 12 12	Slot 12 9	Slot 12 11	Slot 12 11	Slot 12 12
Beacon CC, Staffs - 1st XI 12	Aldridge CC - 1st XI 12	Bloxwich CC - 1st XI 12	Aldridge CC - 2nd XI 12	Bloxwich CC - 2nd XI 12	🟩 Armitage CC - 1st XI 12	Beacon CC, Staffs - 4th XI 12	Cannock CC - 4th XI 12
Brewood CC - 1st XI 12	Cannock CC - 1st XI 12	Brewood CC - 2nd XI 12	Beacon CC, Staffs - 2nd XI 12	Brewood CC - 3rd XI 6	Beacon CC, Staffs - 3rd XI 12	🟩 Bloxwich CC - 3rd XI 12	Coseley CC - 3rd XI 12
🟩 Bye 12	Fordhouses CC - 2nd XI 12	Cannock CC - 2nd XI 12	Hammerwich CC - Sat 2nd XI 12	Burntwood St Matthew's CC 12	Cannock CC - 3rd XI 12	Brewood CC - 4th XI 6	Fordhouses CC - 4th XI 12
Fordhouses CC - 1st XI 12	Penkridge CC - 2nd XI 12	Coseley CC - 1st XI 12	Highcroft and Great Barr CC 12	Codsall CC - 1st XI 6	Coseley CC - 2nd XI 12	Codsall CC - 2nd XI 6	Hammerwich CC - Sat 4th XI 12
Hammerwich CC - Sat 1st XI 12	Rugeley CC - 1st XI 12	Highcroft and Great Barr CC 12	Old Wulfrunians Tettenhall C 12	Fordhouses CC - 3rd XI 12	🟩 Elford CC - 1st XI 12	Hammerwich CC - Sat 3rd XI 12	Old Wulfrunians Tettenhall C 12
Lichfield CC - Saturday 1st X 12	Tamworth CC - 2nd XI 12	Lichfield CC - Saturday 2nd X 12	🟩 Penn CC - 2nd XI 12	Milford Hall CC - 3rd XI 12	🟩 Lichfield CC - Saturday 3rd X 12	Hawkins Sports CC - 1st XI 12	Penkridge CC - 4th XI 12
Milford Hall CC - 1st XI 12	🟥 Walsall CC - 2nd XI 1	Milford Hall CC - 2nd XI 12	Rugeley CC - 2nd XI 12	🟥 Penkridge CC - 3rd XI 1	Old Wulfrunians Tettenhall C 12	Milford Hall CC - 4th XI 12	Rugeley CC - 3rd XI 12
Old Wulfrunians Tettenhall C 12	Wednesbury CC - 1st XI 12	🟥 Pelsall CC - 2nd XI 1	Springhill CC - 1st XI 12	Rushall CC - 1st XI 12	Springvale CC, Staffs - 2nd X 12	Pelsall CC - 3rd XI 12	Rushall CC - 2nd XI 12
Penkridge CC - 1st XI 12	🟥 West Bromwich Dartmouth C 1	Springvale CC, Staffs - 1st XI 12	Tamworth CC - 3rd XI 12	Swindon CC, Staffs - 2nd XI 12	🟩 Walsall CC - 3rd XI 12	Springhill CC - 2nd XI 12	🟩 Swindon CC, Staffs - 3rd XI 12
Tamworth CC - 1st XI 12	Whittington CC, Staffs - Sat 1 12	Swindon CC, Staffs - 1st XI 12	Walsall YPF CC - 2nd XI 12	Whitmore Reans CC - 1st XI 12	Wednesbury CC - 2nd XI 12	🟥 Whittington CC, Staffs - Sat 3 1	Tamworth CC - 4th XI 12
🟥 Wolverhampton CC - 2nd XI 1	Wightwick & Finchfield CC - 12	Walsall YPF CC - 1st XI 12	West Bromwich Dartmouth C 12	Wightwick & Finchfield CC - 12	Whitmore Reans CC - 2nd XI 12	Wightwick & Finchfield CC - 12	West Bromwich Dartmouth C 12
Wombourne CC - 1st XI 12	Wombourne CC - 2nd XI 12	Wolverhampton CC - 3rd XI 12	Whittington CC, Staffs - Sat 2 12	Wolverhampton CC - 4th XI 12	🟥 Wombourne CC - 3rd XI 1	Wolverhampton CC - 5th XI 12	🟩 Wightwick & Finchfield CC - 12

Here colour coding to used help with any manual placement exercise.

- Green indicates a 'golden team', no ground sharing and no issues with needing specific fixtures.
- Red is a highly constrained team limited to fitting into only 1 slot.

It would be desirable to take away the manual placement option for any team which must be in a particular slot (the red teams) and place these at initialisation of the grid. In the current tool a 'one click' button will place all the teams on red and where automated procession is used these are placed before any other teams are added.

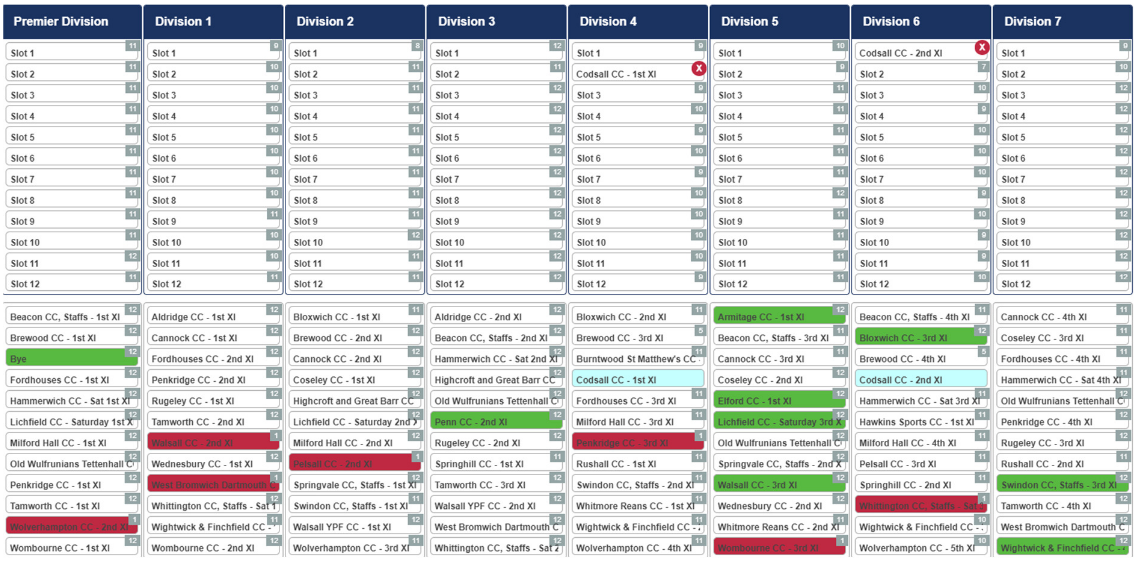
Rolling a mouse over a particular team indicated any team in the same group (blue) and which of the remaining slots the team can fit into (yellow).



Premier Division	Division 1	Division 2	Division 3	Division 4	Division 5	Division 6	Division 7
Slot 1 11	Slot 1 10	Slot 1 11	Slot 1 12	Slot 1 9	Slot 1 11	Slot 1 11	Slot 1 12
Slot 2 11	Slot 2 10	Slot 2 11	Slot 2 12	🟨 Slot 2 11	Slot 2 11	Slot 2 9	Slot 2 12
Slot 3 11	Slot 3 10	Slot 3 11	Slot 3 12	Slot 3 9	Slot 3 11	Slot 3 11	Slot 3 12
Slot 4 11	Slot 4 10	Slot 4 11	Slot 4 12	🟨 Slot 4 11	Slot 4 11	Slot 4 9	Slot 4 12
Slot 5 11	Slot 5 10	Slot 5 11	Slot 5 12	🟨 Slot 5 10	Slot 5 11	Slot 5 10	Slot 5 12
Slot 6 11	Slot 6 10	Slot 6 11	Slot 6 12	Slot 6 10	Slot 6 11	Slot 6 11	Slot 6 12
Slot 7 11	Slot 7 10	Slot 7 11	Slot 7 12	Slot 7 9	Slot 7 11	Slot 7 11	Slot 7 12
Slot 8 11	Slot 8 10	Slot 8 11	Slot 8 12	🟨 Slot 8 11	Slot 8 12	Slot 8 9	Slot 8 12
Slot 9 11	Slot 9 11	Slot 9 11	Slot 9 12	🟨 Slot 9 11	Slot 9 11	Slot 9 10	Slot 9 12
Slot 10 11	Slot 10 10	Slot 10 12	Slot 10 12	Slot 10 10	Slot 10 11	Slot 10 10	Slot 10 12
Slot 11 12	Slot 11 10	Slot 11 11	Slot 11 12	🟨 Slot 11 11	Slot 11 11	Slot 11 9	Slot 11 12
Slot 12 11	Slot 12 11	Slot 12 11	Slot 12 12	Slot 12 9	Slot 12 11	Slot 12 11	Slot 12 12
Beacon CC, Staffs - 1st XI 12	Aldridge CC - 1st XI 12	Bloxwich CC - 1st XI 12	Aldridge CC - 2nd XI 12	Bloxwich CC - 2nd XI 12	🟩 Armitage CC - 1st XI 12	Beacon CC, Staffs - 4th XI 12	Cannock CC - 4th XI 12
Brewood CC - 1st XI 12	Cannock CC - 1st XI 12	Brewood CC - 2nd XI 12	Beacon CC, Staffs - 2nd XI 12	Brewood CC - 3rd XI 6	Beacon CC, Staffs - 3rd XI 12	🟩 Bloxwich CC - 3rd XI 12	Coseley CC - 3rd XI 12
🟩 Bye 12	Fordhouses CC - 2nd XI 12	Cannock CC - 2nd XI 12	Hammerwich CC - Sat 2nd XI 12	Burntwood St Matthew's CC 12	Cannock CC - 3rd XI 12	Brewood CC - 4th XI 6	Fordhouses CC - 4th XI 12
Fordhouses CC - 1st XI 12	Penkridge CC - 2nd XI 12	Coseley CC - 1st XI 12	Highcroft and Great Barr CC 12	🟦 Codsall CC - 1st XI 6	Coseley CC - 2nd XI 12	🟦 Codsall CC - 2nd XI 6	Hammerwich CC - Sat 4th XI 12
Hammerwich CC - Sat 1st XI 12	Rugeley CC - 1st XI 12	Highcroft and Great Barr CC 12	Old Wulfrunians Tettenhall C 12	Fordhouses CC - 3rd XI 12	🟩 Elford CC - 1st XI 12	Hammerwich CC - Sat 3rd XI 12	Old Wulfrunians Tettenhall C 12
Lichfield CC - Saturday 1st X 12	Tamworth CC - 2nd XI 12	Lichfield CC - Saturday 2nd X 12	🟩 Penn CC - 2nd XI 12	Milford Hall CC - 3rd XI 12	🟩 Lichfield CC - Saturday 3rd X 12	Hawkins Sports CC - 1st XI 12	Penkridge CC - 4th XI 12
Milford Hall CC - 1st XI 12	🟥 Walsall CC - 2nd XI 1	Milford Hall CC - 2nd XI 12	Rugeley CC - 2nd XI 12	🟥 Penkridge CC - 3rd XI 1	Old Wulfrunians Tettenhall C 12	Milford Hall CC - 4th XI 12	Rugeley CC - 3rd XI 12
Old Wulfrunians Tettenhall C 12	Wednesbury CC - 1st XI 12	🟥 Pelsall CC - 2nd XI 1	Springhill CC - 1st XI 12	Rushall CC - 1st XI 12	Springvale CC, Staffs - 2nd X 12	Pelsall CC - 3rd XI 12	Rushall CC - 2nd XI 12
Penkridge CC - 1st XI 12	🟥 West Bromwich Dartmouth C 1	Springvale CC, Staffs - 1st XI 12	Tamworth CC - 3rd XI 12	Swindon CC, Staffs - 2nd XI 12	🟩 Walsall CC - 3rd XI 12	Springhill CC - 2nd XI 12	🟩 Swindon CC, Staffs - 3rd XI 12
Tamworth CC - 1st XI 12	Whittington CC, Staffs - Sat 1 12	Swindon CC, Staffs - 1st XI 12	Walsall YPF CC - 2nd XI 12	Whitmore Reans CC - 1st XI 12	Wednesbury CC - 2nd XI 12	🟥 Whittington CC, Staffs - Sat 3 1	Tamworth CC - 4th XI 12
🟥 Wolverhampton CC - 2nd XI 1	Wightwick & Finchfield CC - 12	Walsall YPF CC - 1st XI 12	West Bromwich Dartmouth C 12	Wightwick & Finchfield CC - 12	Whitmore Reans CC - 2nd XI 12	Wightwick & Finchfield CC - 12	West Bromwich Dartmouth C 12
Wombourne CC - 1st XI 12	Wombourne CC - 2nd XI 12	Wolverhampton CC - 3rd XI 12	Whittington CC, Staffs - Sat 2 12	Wolverhampton CC - 4th XI 12	🟥 Wombourne CC - 3rd XI 1	Wolverhampton CC - 5th XI 12	🟩 Wightwick & Finchfield CC - 12

Codsall CC here shares 1st and 2nds on the same group, hovering over the 1st XI highlights blue the second XI and yellow where the 1st XI can occupy.

Dragging Codsall 1st into slot 2 places Codsall 2nds into Slot 1

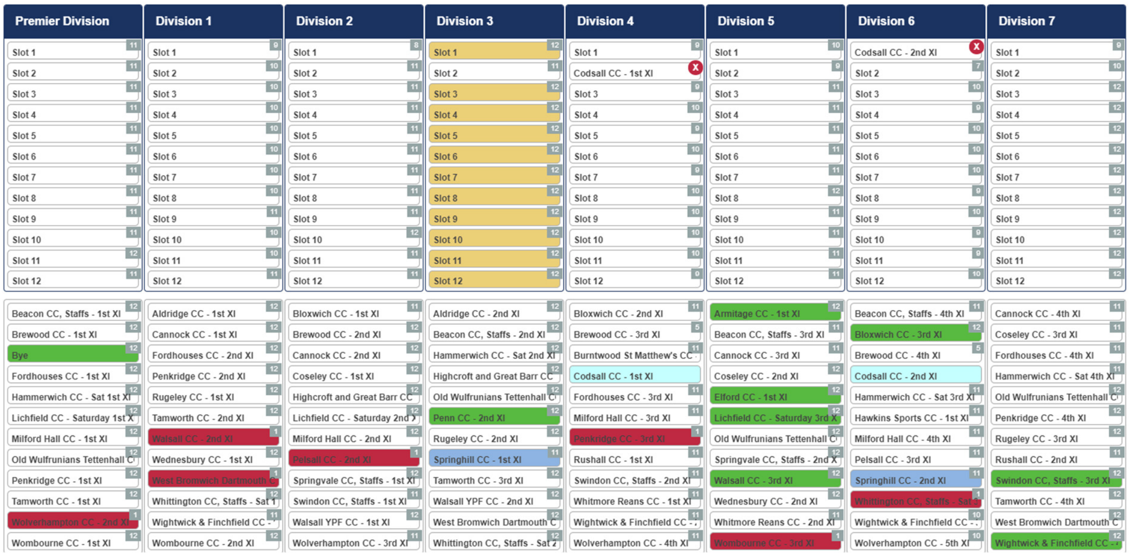


Premier Division	Division 1	Division 2	Division 3	Division 4	Division 5	Division 6	Division 7
Slot 1 11	Slot 1 9	Slot 1 8	Slot 1 12	Slot 1 9	Slot 1 10	Codsall CC - 2nd XI X	Slot 1 9
Slot 2 11	Slot 2 10	Slot 2 11	Slot 2 11	Codsall CC - 1st XI X	Slot 2 9	Slot 2 7	Slot 2 10
Slot 3 11	Slot 3 10	Slot 3 11	Slot 3 12	Slot 3 9	Slot 3 11	Slot 3 10	Slot 3 12
Slot 4 11	Slot 4 10	Slot 4 11	Slot 4 12	Slot 4 10	Slot 4 11	Slot 4 9	Slot 4 12
Slot 5 11	Slot 5 10	Slot 5 11	Slot 5 12	Slot 5 9	Slot 5 11	Slot 5 10	Slot 5 12
Slot 6 11	Slot 6 10	Slot 6 11	Slot 6 12	Slot 6 10	Slot 6 11	Slot 6 10	Slot 6 12
Slot 7 11	Slot 7 10	Slot 7 11	Slot 7 12	Slot 7 9	Slot 7 11	Slot 7 10	Slot 7 12
Slot 8 11	Slot 8 10	Slot 8 11	Slot 8 12	Slot 8 10	Slot 8 12	Slot 8 9	Slot 8 12
Slot 9 11	Slot 9 11	Slot 9 11	Slot 9 12	Slot 9 10	Slot 9 11	Slot 9 10	Slot 9 12
Slot 10 11	Slot 10 10	Slot 10 12	Slot 10 12	Slot 10 10	Slot 10 11	Slot 10 9	Slot 10 12
Slot 11 12	Slot 11 10	Slot 11 11	Slot 11 12	Slot 11 10	Slot 11 11	Slot 11 9	Slot 11 12
Slot 12 11	Slot 12 11	Slot 12 11	Slot 12 12	Slot 12 9	Slot 12 11	Slot 12 10	Slot 12 12
Beacon CC, Staffs - 1st XI 12	Aldridge CC - 1st XI 12	Bloxwich CC - 1st XI 11	Aldridge CC - 2nd XI 12	Bloxwich CC - 2nd XI 11	Armitage CC - 1st XI 12	Beacon CC, Staffs - 4th XI 11	Cannock CC - 4th XI 11
Brewood CC - 1st XI 12	Cannock CC - 1st XI 12	Brewood CC - 2nd XI 12	Beacon CC, Staffs - 2nd XI 12	Brewood CC - 3rd XI 5	Beacon CC, Staffs - 3rd XI 11	Bloxwich CC - 3rd XI 12	Coseley CC - 3rd XI 11
Bye 12	Fordhouses CC - 2nd XI 12	Cannock CC - 2nd XI 12	Hammerwich CC - Sat 2nd XI 12	Burntwood St Matthew's CC 11	Cannock CC - 3rd XI 11	Brewood CC - 4th XI 5	Fordhouses CC - 4th XI 11
Fordhouses CC - 1st XI 12	Penkridge CC - 2nd XI 12	Coseley CC - 1st XI 12	Highcroft and Great Barr CC 12	Codsall CC - 1st XI	Coseley CC - 2nd XI 12	Codsall CC - 2nd XI	Hammerwich CC - Sat 4th XI 11
Hammerwich CC - Sat 1st XI 12	Rugeley CC - 1st XI 12	Highcroft and Great Barr CC 12	Old Wulfrunians Tettenhall C 12	Fordhouses CC - 3rd XI 11	Elford CC - 1st XI 12	Hammerwich CC - Sat 3rd XI 11	Old Wulfrunians Tettenhall C 12
Lichfield CC - Saturday 1st X 12	Tamworth CC - 2nd XI 12	Lichfield CC - Saturday 2nd X 12	Penn CC - 2nd XI 12	Milford Hall CC - 3rd XI 11	Lichfield CC - Saturday 3rd X 12	Hawkins Sports CC - 1st XI 11	Penkridge CC - 4th XI 12
Milford Hall CC - 1st XI 12	Walsall CC - 2nd XI 1	Milford Hall CC - 2nd XI 12	Rugeley CC - 2nd XI 12	Penkridge CC - 3rd XI 1	Old Wulfrunians Tettenhall C 12	Milford Hall CC - 4th XI 11	Rugeley CC - 3rd XI 12
Old Wulfrunians Tettenhall C 12	Wednesbury CC - 1st XI 12	Pelsall CC - 2nd XI 1	Springhill CC - 1st XI 11	Rushall CC - 1st XI 11	Springvale CC, Staffs - 2nd X 12	Pelsall CC - 3rd XI 11	Rushall CC - 2nd XI 11
Penkridge CC - 1st XI 12	West Bromwich Dartmouth C 1	Springvale CC, Staffs - 1st XI 12	Tamworth CC - 3rd XI 12	Swindon CC, Staffs - 2nd XI 11	Walsall CC - 3rd XI 12	Springhill CC - 2nd XI 11	Swindon CC, Staffs - 3rd XI 12
Tamworth CC - 1st XI 12	Whittington CC, Staffs - Sat 1 12	Swindon CC, Staffs - 1st XI 11	Walsall YPF CC - 2nd XI 12	Whitmore Reans CC - 1st XI 11	Wednesbury CC - 2nd XI 12	Whittington CC, Staffs - Sat 3 1	Tamworth CC - 4th XI 12
Wolverhampton CC - 2nd XI 1	Wightwick & Finchfield CC - 11	Walsall YPF CC - 1st XI 12	West Bromwich Dartmouth C 12	Wightwick & Finchfield CC - 11	Whitmore Reans CC - 2nd XI 11	Wightwick & Finchfield CC - 10	West Bromwich Dartmouth C 12
Wombourne CC - 1st XI 12	Wombourne CC - 2nd XI 12	Wolverhampton CC - 3rd XI 11	Whittington CC, Staffs - Sat 2 12	Wolverhampton CC - 4th XI 11	Wombourne CC - 3rd XI 1	Wolverhampton CC - 5th XI 10	Wightwick & Finchfield CC - 12

A remove button appears to allow the operation to be reversed. There should also be an undo function.

Having placed a team some indication is needed that it has been actioned, here it is to colour the team light blue in the bottom grid and make in non draggable.

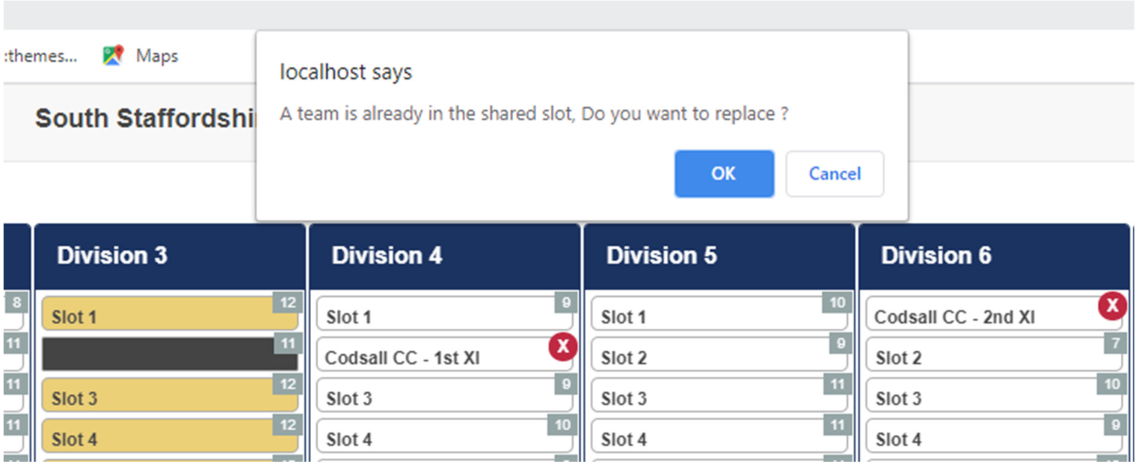
Highlighting Springfield 1sts then shows



Premier Division	Division 1	Division 2	Division 3	Division 4	Division 5	Division 6	Division 7
Slot 1 11	Slot 1 9	Slot 1 8	Slot 1 12	Slot 1 9	Slot 1 10	Codsall CC - 2nd XI X	Slot 1 9
Slot 2 11	Slot 2 10	Slot 2 11	Slot 2 11	Codsall CC - 1st XI X	Slot 2 9	Slot 2 7	Slot 2 10
Slot 3 11	Slot 3 10	Slot 3 11	Slot 3 12	Slot 3 9	Slot 3 11	Slot 3 10	Slot 3 12
Slot 4 11	Slot 4 10	Slot 4 11	Slot 4 12	Slot 4 10	Slot 4 11	Slot 4 9	Slot 4 12
Slot 5 11	Slot 5 10	Slot 5 11	Slot 5 12	Slot 5 9	Slot 5 11	Slot 5 10	Slot 5 12
Slot 6 11	Slot 6 10	Slot 6 11	Slot 6 12	Slot 6 10	Slot 6 11	Slot 6 10	Slot 6 12
Slot 7 11	Slot 7 10	Slot 7 11	Slot 7 12	Slot 7 9	Slot 7 11	Slot 7 10	Slot 7 12
Slot 8 11	Slot 8 10	Slot 8 11	Slot 8 12	Slot 8 10	Slot 8 12	Slot 8 9	Slot 8 12
Slot 9 11	Slot 9 11	Slot 9 11	Slot 9 12	Slot 9 10	Slot 9 11	Slot 9 10	Slot 9 12
Slot 10 11	Slot 10 10	Slot 10 12	Slot 10 12	Slot 10 10	Slot 10 11	Slot 10 9	Slot 10 12
Slot 11 12	Slot 11 10	Slot 11 11	Slot 11 12	Slot 11 10	Slot 11 11	Slot 11 9	Slot 11 12
Slot 12 11	Slot 12 11	Slot 12 11	Slot 12 12	Slot 12 9	Slot 12 11	Slot 12 10	Slot 12 12
Beacon CC, Staffs - 1st XI 12	Aldridge CC - 1st XI 12	Bloxwich CC - 1st XI 11	Aldridge CC - 2nd XI 12	Bloxwich CC - 2nd XI 11	Armitage CC - 1st XI 12	Beacon CC, Staffs - 4th XI 11	Cannock CC - 4th XI 11
Brewood CC - 1st XI 12	Cannock CC - 1st XI 12	Brewood CC - 2nd XI 12	Beacon CC, Staffs - 2nd XI 12	Brewood CC - 3rd XI 5	Beacon CC, Staffs - 3rd XI 11	Bloxwich CC - 3rd XI 12	Coseley CC - 3rd XI 11
Bye 12	Fordhouses CC - 2nd XI 12	Cannock CC - 2nd XI 12	Hammerwich CC - Sat 2nd XI 12	Burntwood St Matthew's CC 11	Cannock CC - 3rd XI 11	Brewood CC - 4th XI 5	Fordhouses CC - 4th XI 11
Fordhouses CC - 1st XI 12	Penkridge CC - 2nd XI 12	Coseley CC - 1st XI 12	Highcroft and Great Barr CC 12	Codsall CC - 1st XI	Coseley CC - 2nd XI 12	Codsall CC - 2nd XI	Hammerwich CC - Sat 4th XI 11
Hammerwich CC - Sat 1st XI 12	Rugeley CC - 1st XI 12	Highcroft and Great Barr CC 12	Old Wulfrunians Tettenhall C 12	Fordhouses CC - 3rd XI 11	Elford CC - 1st XI 12	Hammerwich CC - Sat 3rd XI 11	Old Wulfrunians Tettenhall C 12
Lichfield CC - Saturday 1st X 12	Tamworth CC - 2nd XI 12	Lichfield CC - Saturday 2nd X 12	Penn CC - 2nd XI 12	Milford Hall CC - 3rd XI 11	Lichfield CC - Saturday 3rd X 12	Hawkins Sports CC - 1st XI 11	Penkridge CC - 4th XI 12
Milford Hall CC - 1st XI 12	Walsall CC - 2nd XI 1	Milford Hall CC - 2nd XI 12	Rugeley CC - 2nd XI 12	Penkridge CC - 3rd XI 1	Old Wulfrunians Tettenhall C 12	Milford Hall CC - 4th XI 11	Rugeley CC - 3rd XI 12
Old Wulfrunians Tettenhall C 12	Wednesbury CC - 1st XI 12	Pelsall CC - 2nd XI 1	Springhill CC - 1st XI 11	Rushall CC - 1st XI 11	Springvale CC, Staffs - 2nd X 12	Pelsall CC - 3rd XI 11	Rushall CC - 2nd XI 11
Penkridge CC - 1st XI 12	West Bromwich Dartmouth C 1	Springvale CC, Staffs - 1st XI 12	Tamworth CC - 3rd XI 12	Swindon CC, Staffs - 2nd XI 11	Walsall CC - 3rd XI 12	Springhill CC - 2nd XI 11	Swindon CC, Staffs - 3rd XI 12
Tamworth CC - 1st XI 12	Whittington CC, Staffs - Sat 1 12	Swindon CC, Staffs - 1st XI 11	Walsall YPF CC - 2nd XI 12	Whitmore Reans CC - 1st XI 11	Wednesbury CC - 2nd XI 12	Whittington CC, Staffs - Sat 3 1	Tamworth CC - 4th XI 12
Wolverhampton CC - 2nd XI 1	Wightwick & Finchfield CC - 11	Walsall YPF CC - 1st XI 12	West Bromwich Dartmouth C 12	Wightwick & Finchfield CC - 11	Whitmore Reans CC - 2nd XI 11	Wightwick & Finchfield CC - 10	West Bromwich Dartmouth C 12
Wombourne CC - 1st XI 12	Wombourne CC - 2nd XI 12	Wolverhampton CC - 3rd XI 11	Whittington CC, Staffs - Sat 2 12	Wolverhampton CC - 4th XI 11	Wombourne CC - 3rd XI 1	Wolverhampton CC - 5th XI 10	Wightwick & Finchfield CC - 12

Springfield 2nds is again highlighted and all slots excluding 2 show available. This is because Codsall already occupy slot 1 in the same division as Springfield 2nds.

This does not stop us trying to place Springfield 1sts into Slot2 and if we do the fixtures admin is suitably warned. The principle should be the users can override the restriction by ignoring the warnings.



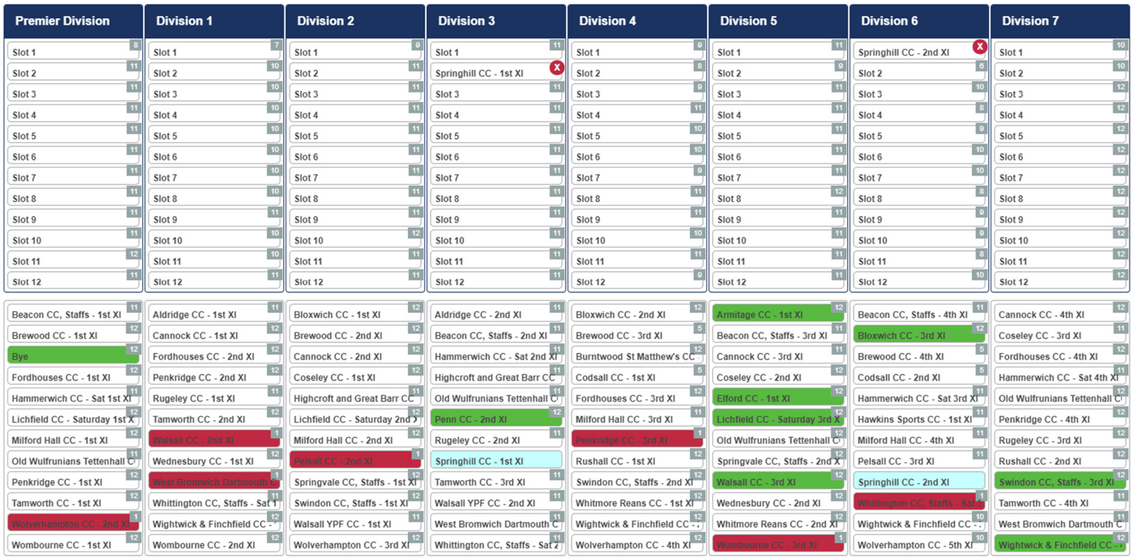
South Staffordshi

localhost says

A team is already in the shared slot, Do you want to replace ?

	Division 3	Division 4	Division 5	Division 6
8	Slot 1 12	Slot 1 9	Slot 1 10	Codsall CC - 2nd XI ☒
11	11	Codsall CC - 1st XI ☒	Slot 2 9	Slot 2 7
11	Slot 3 12	Slot 3 9	Slot 3 11	Slot 3 10
11	Slot 4 12	Slot 4 10	Slot 4 11	Slot 4 9

Pressing OK will replace Codsall 2nds with Springfield 2nds and remove Codsall 1sts



Premier Division	Division 1	Division 2	Division 3	Division 4	Division 5	Division 6	Division 7
Slot 1 8	Slot 1 7	Slot 1 9	Slot 1 11	Slot 1 9	Slot 1 11	Springhill CC - 2nd XI ☒	Slot 1 10
Slot 2 11	Slot 2 10	Slot 2 11	Springhill CC - 1st XI ☒	Slot 2 8	Slot 2 9	Slot 2 6	Slot 2 10
Slot 3 11	Slot 3 10	Slot 3 11	Slot 3 11	Slot 3 9	Slot 3 11	Slot 3 10	Slot 3 12
Slot 4 11	Slot 4 10	Slot 4 11	Slot 4 11	Slot 4 11	Slot 4 11	Slot 4 8	Slot 4 12
Slot 5 11	Slot 5 10	Slot 5 11	Slot 5 11	Slot 5 10	Slot 5 11	Slot 5 9	Slot 5 12
Slot 6 11	Slot 6 10	Slot 6 11	Slot 6 11	Slot 6 10	Slot 6 11	Slot 6 10	Slot 6 12
Slot 7 11	Slot 7 10	Slot 7 11	Slot 7 11	Slot 7 9	Slot 7 11	Slot 7 10	Slot 7 12
Slot 8 11	Slot 8 10	Slot 8 11	Slot 8 11	Slot 8 11	Slot 8 12	Slot 8 8	Slot 8 12
Slot 9 11	Slot 9 11	Slot 9 11	Slot 9 11	Slot 9 11	Slot 9 11	Slot 9 9	Slot 9 12
Slot 10 11	Slot 10 10	Slot 10 12	Slot 10 11	Slot 10 10	Slot 10 11	Slot 10 9	Slot 10 12
Slot 11 12	Slot 11 10	Slot 11 11	Slot 11 11	Slot 11 11	Slot 11 11	Slot 11 8	Slot 11 12
Slot 12 11	Slot 12 11	Slot 12 11	Slot 12 11	Slot 12 9	Slot 12 11	Slot 12 10	Slot 12 12

Beacon CC, Staffs - 1st XI 11	Aldridge CC - 1st XI 11	Bloxwich CC - 1st XI 12	Aldridge CC - 2nd XI 11	Bloxwich CC - 2nd XI 12	Armitage CC - 1st XI 12	Beacon CC, Staffs - 4th XI 11	Cannock CC - 4th XI 12
Brewood CC - 1st XI 12	Cannock CC - 1st XI 12	Brewood CC - 2nd XI 12	Beacon CC, Staffs - 2nd XI 11	Brewood CC - 3rd XI 5	Beacon CC, Staffs - 3rd XI 11	Bloxwich CC - 3rd XI 12	Coseley CC - 3rd XI 11
Bye 12	Fordhouses CC - 2nd XI 12	Cannock CC - 2nd XI 12	Hammerwich CC - Sat 2nd XI 11	Burntwood St Matthew's CC 12	Cannock CC - 3rd XI 11	Brewood CC - 4th XI 5	Fordhouses CC - 4th XI 12
Fordhouses CC - 1st XI 12	Penkridge CC - 2nd XI 12	Coseley CC - 1st XI 12	Highcroft and Great Barr CC 11	Codsall CC - 1st XI 5	Coseley CC - 2nd XI 12	Codsall CC - 2nd XI 5	Hammerwich CC - Sat 4th XI 11
Hammerwich CC - Sat 1st XI 11	Rugeley CC - 1st XI 11	Highcroft and Great Barr CC 11	Old Wulfrunians Tettenhall C 11	Fordhouses CC - 3rd XI 12	Elford CC - 1st XI 12	Hammerwich CC - Sat 3rd XI 11	Old Wulfrunians Tettenhall C 12
Lichfield CC - Saturday 1st X 12	Tamworth CC - 2nd XI 12	Lichfield CC - Saturday 2nd X 12	Penn CC - 2nd XI 12	Milford Hall CC - 3rd XI 11	Lichfield CC - Saturday 3rd X 12	Hawkins Sports CC - 1st XI 11	Penkridge CC - 4th XI 12
Milford Hall CC - 1st XI 12	Walsall CC - 2nd XI 1	Milford Hall CC - 2nd XI 12	Rugeley CC - 2nd XI 11	Penkridge CC - 3rd XI 1	Old Wulfrunians Tettenhall C 12	Milford Hall CC - 4th XI 11	Rugeley CC - 3rd XI 12
Old Wulfrunians Tettenhall C 11	Wednesbury CC - 1st XI 12	Pelsall CC - 2nd XI 1	Springhill CC - 1st XI	Rushall CC - 1st XI 12	Springvale CC, Staffs - 2nd X 12	Pelsall CC - 3rd XI 11	Rushall CC - 2nd XI 12
Penkridge CC - 1st XI 12	West Bromwich Dartmouth C 1	Springvale CC, Staffs - 1st XI 12	Tamworth CC - 3rd XI 11	Swindon CC, Staffs - 2nd XI 12	Walsall CC - 3rd XI 12	Springhill CC - 2nd XI	Swindon CC, Staffs - 3rd XI 12
Tamworth CC - 1st XI 12	Whittington CC, Staffs - Sat 1 11	Swindon CC, Staffs - 1st XI 12	Walsall YPF CC - 2nd XI 11	Whitmore Reans CC - 1st XI 12	Wednesbury CC - 2nd XI 12	Whittington CC, Staffs - Sat 3 1	Tamworth CC - 4th XI 11
Wolverhampton CC - 2nd XI 1	Wightwick & Finchfield CC - 12	Walsall YPF CC - 1st XI 11	West Bromwich Dartmouth C 11	Wightwick & Finchfield CC - 12	Whitmore Reans CC - 2nd XI 12	Wightwick & Finchfield CC - 10	West Bromwich Dartmouth C 11
Wombourne CC - 1st XI 12	Wombourne CC - 2nd XI 12	Wolverhampton CC - 3rd XI 12	Whittington CC, Staffs - Sat 2 11	Wolverhampton CC - 4th XI 12	Wombourne CC - 3rd XI 1	Wolverhampton CC - 5th XI 10	Wightwick & Finchfield CC - 12

Eventually this way the grid can be completed such that all the teams in the bottom of the screen are allocated to positions at the top of the screen.

If after a team is placed it means another team has no viable positions the fixture administrator should be alerted so the y can immediately resolve the issue rather than placing more teams before they notice the issue.

Automated allocations

To understand the possibility for automatically completing the grid we need to examine the thought processes around completing the top grid.

There are some constraints which are absolute givens and therefore can be applied first. Any team in red on this grid can only fit in one place therefore the person would know to place these first so they did not inadvertently use that slot for a team which could fit in multiple places.

The teams in green can go anywhere with no constraints so placing these last is the best option.

As for the rest it is difficult to determine a best general approach, sometimes working top left to bottom right seems best, other times completing row 1 and therefore row 2 then moving to row 3 and 4 might be better. Sometimes placing the teams with the highest number of constraints seems to work best.

Making such determination is a case of a bit of experience and a gut feel for the problem and hard to program without a decree of AI in the logic.

Possible approaches

The sample tool takes 2 possible approaches. Whilst either of these can be used these are very much a start point and alternative approaches which complete the grid in an efficient way for multiple problems is acceptable.

Approach 1 - Least moves approach

The approach is to first is to determine which team has the least number of possible moves, place that team, re-evaluate which team should be next to place as placing 1 team affects the ability to place the next, place the next team, re-evaluate and so on.

Whilst this sometimes works some difficulties arise when the method hits a dead end, the next team to place has no available slots.

The person in these circumstances would look at moving an already placed group of teams to make the next team fit.

The manual approach also might be more complex then just changing 1 possible set of teams looking for 3 or more teams to move. Ultimately if all looks lost the person might remove all or a majority of teams and try a different approach.

This semi random backtracking based in some part on experience and in some part informed guess work is hard to program.

Because there is a degree of randomness and learning in this approach it is also very easy to get into a loop of trying the same thing over and over again, a person recognises that the same combinations are coming up over and over again and resolves these from experience.

An automated approach using the same techniques might be possible but guarding against getting into a loop and tracking all the combinations tried to date has proved difficult. To store all the tried combinations and recognise these in a program would be resource intensive.

Also if on every move the next best move is revaluated then looking ahead several moves as in a chess problem would seem a good approach but this has to be balanced against the processing time and number of combinations each move takes.

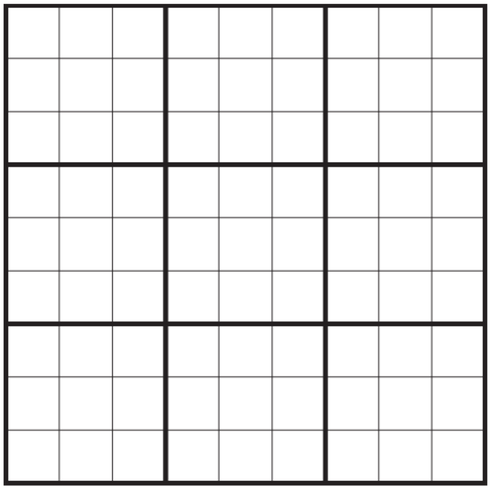
That said a future enhancement could be to take this approach of placing the team with the least amount of leeway as far as it will go before changing tactics to approach 2.

If there are better approaches out there then we are open to these being discussed and demonstrated.

Approach 2- Trial and backtrack approach

The issue of placing the teams in a grid shares some similarities with other mathematical problems.

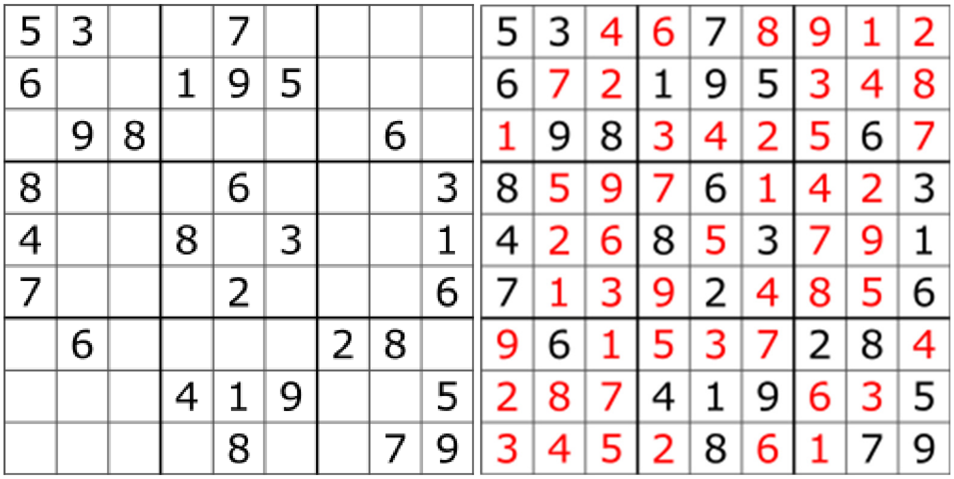
For instance, in sudoku numbers are placed in several squares made up of 3 by 3 smaller squares. The overall sudoku grid is of 9 by 9 squares, divided into 9 sets of 3 by 3 squares.



The placement of the numbers 1 to 9 in these squares within a square in contingent on where number are placed in the other squares.

A start point is given were some of the squares contain numbers and the objective is to find an answer which places the other numbers. In sudoku the creators of the puzzle is trying to arrange the start point such that there is only 1 possible solution and that solution is not trivial to find.

A typical start point and solution is (30 numbers were given 51 deduced in solving the puzzle), there being only 1 solution to the problem (I think).



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

If a sudoku player was given a grid of 9 by 9 squares with no number prepopulated there would be many different solutions. In fact isn't this the way a sudoku puzzle is created, fill in a blank grid then take most of the numbers out and have the solver work out where to put then back in ?.

With a bit of imagination if we took each of the 9 squares or 9 and rearranged them into a table of columns and rows we would get.

	SQ1	SQ2	SQ3	SQ4	SQ5	SQ6	SQ7	SQ8	SQ9
Pos 1	5			8					2
Pos 2	3	7			6		6		8
Pos 3						3			
Pos 4	6	1		4	8			4	
Pos 5		9						1	
Pos 6		5			3	1		9	5
Pos 7				7					
Pos 8	9		6		2			8	7
Pos 9	8					6			9

This is not dissimilar to the placement of teams into a grid in the fixture generation problem. If we looked at the grid above and had to place the number 2 in column SQ1 there would be limits (similar to ground sharing group) as to where number 2 can fit, and there would be squares the number 2 could not fit as they are already occupied.

The cracking of Sudoku 9 by 9 grid problems has many articles written on 'how to', the problems are similar to fixture generation from templates therefore can the same approach be used ?.

https://en.wikipedia.org/wiki/Sudoku_solving_algorithms

which shows such a puzzle being cracked using a 'Backtracking Algorithm'

As allocation of the group of teams to our grid has some similarities to the Sudoku puzzle potentially using a backtracking algorithm could resolve automatically our fixture generation problem.

Whilst the problems have similarities there are 2 big differences between a sudoku problem and generation of fixtures.

1) Sudoku is contrived to have 1 solution based on the initial constraints; the fixture generation problem may have many solutions which all work.
2) The Sudoku problem is taken on trust as solvable, there may be fixture generation problems which are so highly constrained are not solvable.

Fixture generation using backtracking

The logic of a simple backtrack algorithm doing a brute force trial and error approach to the fixture generation problem goes like this

After placing on the teams which can only go in one place.

- Setup an array for the solution.
- Setup an array of groups containing the teams in each group
- Setup up an array of 'golden teams'

Once we are setup ready to go loop through the groups in order

- Place Group A in slot 1, it fits so move to Group B
- Place Group B in slot 1, it does not fit as occupied by Group A
- Place Group B in slot 2, it fits move to team C

Eventually a group might not fit any slot (a dead end)

- Place Group X and try in turn each slot and reach Slot 12 the last possible slot, Group X does not fit Slot 12
- Go back to Group W
- Move Group W up 1 slot
- Now try Group X again.

Such a backtrack algorithm could then find not only Group X but Group W and Group V ceases to have options and backtrack to Group U or even all the way back in theory to Group A.

However, it is working in away that no combination will be duplicated and therefore there will not be an infinite loop and if there is a solution to the problem it will be found and we do not need to check on each iteration if that combination has been tries before.

At the end place the 'golden teams'

The disadvantage of this is the algorithm is trying things which to the human eye are obviously wrong.

The advantage is that there is a logical progression and backtrack which is simple to program and takes very little logic at each iteration of the loop meaning it is possible to try a large number of possibilities quickly.

Therefore, if the algorithm tries 100 silly things to find 1 good move and the 100 tries take a few milliseconds it will outperform the person.

The prototype tool uses this crude but seemingly affective approach. The code is setup to make maximum use of arrays and minimise the number of loops in each iteration looking a speed over intelligence. It is written in JavaScript and runs on the client so takes no server power to run but this approach is not necessarily the best. The code of the algorithm is purposely kept simple in an attempt to make it easy to understand and has not been extensively optimised for performance. That said in testing it achieves about 1 million iterations a second.

Note a refinement could be to increase the checking on each loop to look at if I do this then Is there any team left which can fill the remaining slots in the division if not move on without potentially trying another few hundred moves before getting the same answer by trial and error. Whilst this can be done each check adds complexity to the method and coding and increases each iteration time.

Such attempts to trade loop speed versus check has not been tried.

Any algorithm code is provided on an asis basis and no guaranteed bug free.

Results of limited trials

The largest senior recreational cricket league in England at 35 divisions and 342 clubs is in Sussex. Most teams share grounds and for testing a few restrictions where added at random.

The league is all 10 team divisions so a single template.

Result : a few seconds with the actual placement routine making around 2000 attempts in 9 milliseconds.

Northants; a smaller league of 15 divisions with a structure including 12, 10 and 8 team divisions and 3 different matching templates. Manually with the same list of constraints the fixtures took 10 hours over a few days to get done.

47 million combinations tried and 40 seconds elapsed

Some other tried based on duplicating the 2020 fixtures done manually

- Staffs 207 moves 2 milliseconds
- Shropshire 185 moves 2 milliseconds
- Worcester 124K moves 166 milliseconds
- Warwickshire 577m moves in 6 minutes

The maths of this on the face of it are daunting. The way of placing say 100 groups of teams in order is 100 factorial, 100 x 99 x 98 etc, a lot of zeros very quickly.

If we took the sudoku example it has been established that approximately 5.96 x 11^{26} final grids exist. However most players if they were given an absolutely blank grid could construct one of those vast numbers of solution very quickly. This is what happened when Sussex the largest league was tested. Although vast numbers of combinations were possible there were also vast numbers of solutions.

At the other end of the spectrum if there are vast numbers of constants and shares and teams which must be placed in particular positions but a solution exists, it becomes like the hard Sudoku problem where the number of combinations is reduced dramatically but there is one or a few solutions.

The hard problems are those were there are just enough constraints to limit the possible number of solutions but still enough combinations to try to ensure there are a lot of iterations to find one of the fewer solutions which fits.

Looking at the sample data Northants which took 47 million combinations and Warwickshire with 577 million are problems were there are many different sized divisions and lots of constraints. In the Northants case several teams needed to avoid some positions in the grid because the 10 and 8 team templates did not precisely match, in the Warwickshire case as well as a mix of 12 and 10 team leagues many clubs needed to be placed in particular placings to match the requirements of fixtures done previously for a higher league.

From a testing point of view all apart from the Sussex example had been done manually by the tester, despite the automated approach have most issues with Warwickshire the manual approach found Worcestershire with many requests to avoid matches on specific dates and Northants which has a combination of 12,10 and 8 team leagues harder to solve. This we think is the difficultly manually in previous tools of keeping track of the additional constraints on top of basic ground sharing.

Even when there are a lot of shares then the size of the league does not currently seem the constraining factor. What Northants, Worcester and Warwickshire have in common is a complex mix of divisions and constraints like cannot play at home on this date.

In manual terms Sussex with 35 divisions would take a while even if you knew exactly were each team was to fit as there are about 150 groups to drag into place. Even with no clashes and a lot of luck that is not trivial.

Additional requirements

The current prototype solution is an either of between automated and manual completion.

An enhancement should be

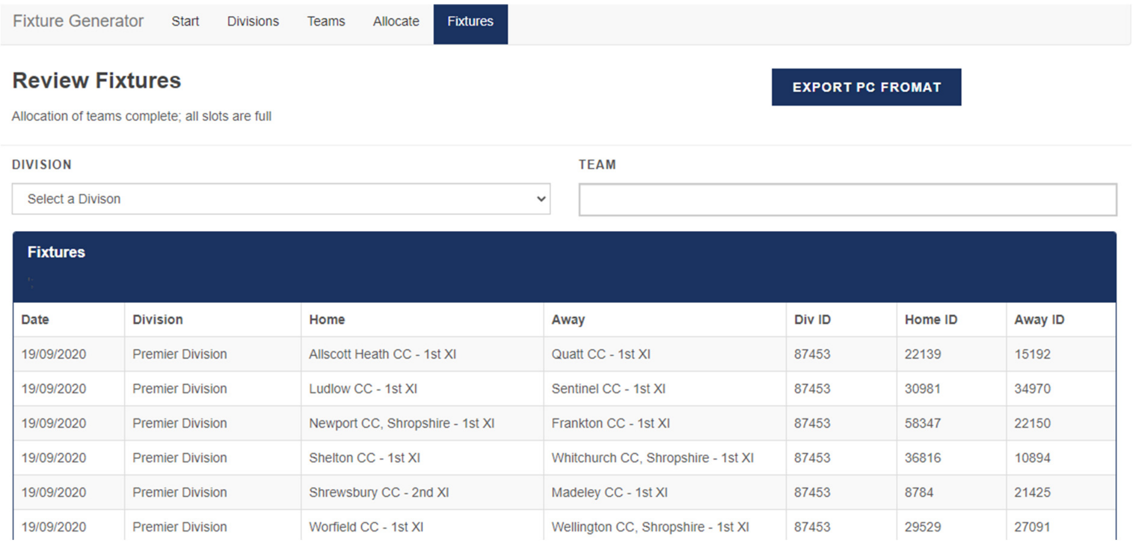
- Allow the user to place a subset of the clubs and then try to automatically add the rest
- Allow the user to take a solution and either add an additional constraint of move a team of teams between division and have the tool try to recalculate the grid

Generating the fixtures

Once the teams are allocated to the grid, generating the fixture is a case of applying the template for each division to the fixtures.

An on screen display must be provided

The example from the prototype tool is



Review Fixtures

Allocation of teams complete; all slots are full

DIVISION

Select a Divison

TEAM

Fixtures

Date	Division	Home	Away	Div ID	Home ID	Away ID
19/09/2020	Premier Division	Allscott Heath CC - 1st XI	Quatt CC - 1st XI	87453	22139	15192
19/09/2020	Premier Division	Ludlow CC - 1st XI	Sentinel CC - 1st XI	87453	30981	34970
19/09/2020	Premier Division	Newport CC, Shropshire - 1st XI	Frankton CC - 1st XI	87453	58347	22150
19/09/2020	Premier Division	Shelton CC - 1st XI	Whitchurch CC, Shropshire - 1st XI	87453	36816	10894
19/09/2020	Premier Division	Shrewsbury CC - 2nd XI	Madeley CC - 1st XI	87453	8784	21425
19/09/2020	Premier Division	Worfield CC - 1st XI	Wellington CC, Shropshire - 1st XI	87453	29529	27091

It should be possible to filter the data by division, club and team.

Note the prototype tool can not filter by club because it is taking as import the play cricket system ids and whilst each team in that export is identified.

The tool should allow export of the fixtures in one or more agreed format. If the tool is standalone to play cricket an export in the format that Play-Cricket can import is essential. Ideally a fully configurable export allowing the user to select the order of column should be provided.

Exports should be able to export all of a selected subset of fixtures.

The generation should show any fixtures which break the constraints given a user can in the allocation phase override the constraints.

The tool should allow the production of a printed fixture list by club, team and division