Each team plays 2 games, 2 different opponents each week, for six weeks. If there is a bye week the schedule needs to extend to 7 weeks to satisfy the requirement of each team playing 12 games.

Back-to-back is preferred, different opponents are prioritized. All games start at 11:00A.M. and the last game starts at 4:00 (if needed).

By-Weeks are possible if needed.

The only "rule" determining bye weeks is field availability.

Each team needs to play each other team an equal amount of times, when possible. (You can have team A playing team B twice, as long as they have played all other teams at least one time already).

User would input the number of fields available for play for each division location.

User would input the teams into the database.

The output would be for the ENTIRE season of six weeks (12 games per team, 7 weeks if there is a bye-week)

There should be a way to update teams or delete teams from the database if necessary.

All teams playing in the same division and season to play a total of 12 games, 2 games per week on the same day (Sunday for all divisions except south county Thursdays)

All kickball fields must be played on at the same time when possible. (All fields have games starting at 11:00am, and all fields would have hourly games thereafter.)

This needs to span for the entirety of the season (6 weeks).

The season will start on the selected season start date. Example: User selects 11/6/2016 as the start date.

Week 1 would occur on the selected date, week 2 will be the following week, week 3 will be the week following week, etc.

**Start-to-Finish step-by-step process:**

Step 1: The user launches the program. (preferably from an executable to avoid having to run visual studio)

Step 2: The user selects from the Top Menu to 'Add Team'

Step 3: The user fills out the add team form consisting of the following: Season, Division Location, Desired Team Name.

Step 4: The user selects to add team, which stores the team in the database. (It would be nice if the user could enter multiple teams at the same time, but not necessary)

Step 5: The user is sent back to the Top Menu after they are finished adding teams.

Step 6: The user selects the Generate Schedule button to be brought to the generate schedule form.

Step 7: The user fills out the generate schedule form consisting of the following variables: Division Location, Season, Season Start Date, and number of fields available for that division.

Step 8: The user now clicks the 'generate schedule' button, creating the schedule for that division and season.

Step 9: The schedule is generated and needs to have the following:

Step 10: The schedule is then saved to the Full Schedule table in the database.

Step 11: The user is sent back to the top menu.

Step 12: The user selects 'view schedule'

Step 13: The user enters the division location, the season, and team name from a combo box consisting of the teams that are playing in that season and division.

Step 14: The user selects 'view schedule' within the view schedule form.

Step 15: The data from the Full Schedule form meeting the credentials selected will be pulled from the Full Schedule form and shown in a PDF format. (The user can just view the full schedule for a division/season, or specifically for a team within the division and season.

Step 16: The user then closes the program.

**Things that cannot happen:**

A team within a division and season cannot be playing against itself.

A team within a division and season cannot be playing on two separate fields at the same time.

A team within a division and season should not play the same team multiple times when other teams have not been played against yet.

Teams cannot play against other teams in separate divisions.

Teams cannot play against other teams in different seasons.

A team cannot play more than 2 games in a week.

A team cannot play more than 12 games per season.

A user of the system should not be able to delete a season in progress.

A user of the system should not be able to create a schedule for a season and division currently in progress.

**Considerations**

**What happens when a power outage happens?**

Currently, the user would lose the data currently in the form if within the 'add team' form, but not the data that has been sent to the database.

**Disaster Recovery:**

The user would back-up the entire program and database on a flash drive.

The schedule should be backed up online when appropriate.

**What needs to happen when a user fills out a field incorrectly?**

The user needs to be prompted, and that data must not be sent inaccurately, the user needs to re-enter the data or perhaps the data should be converted to protect against user-error.

**There needs to be a user manual and training regimen.**