

# Soccer League Board

---

## System Specification:

**OS:** High Sierra

**Ruby:** 2.4.1

**Yard:** 0.9.12

---

## How to run the console-app

**To run the script:**

```
ruby soccer_league_board.rb
```

**To run the test script:**

```
ruby soccer_league_board_test.rb
```

---

## How to see the code documentation YARD

**If you don't have it, install YARD as next:**

```
gem install yard
```

**For generate the doc:**

```
yardoc soccer_league_board.rb
```

**YARD local server for see the documentation in the browser run:**

```
yard server
```

Then go to link:

```
http://localhost:8808
```

---

## Rules

In this league, a draw (tie) is worth 1 point and a win is worth 3 points. A loss is worth 0 points. If two or more teams have the same number of points, they should have the same rank and be printed in alphabetical order

---

## Technical Specifications

1. User will be prompt to input a file\_name
2. If file do not exist in folder, it will prompt to user again that file\_name is incorrect and re-enter file name.
3. When a valid file\_name ia provided, it will read file line by line and parse the data into required data structure to calculate the points for each team.
4. Points will be calculated according to game rules and we will rank the teams based on points. If there is a tie between two points for sae points, it will be displayed in alphabetical order as provided in requirements.
5. The calculated ranks will be displayed on console to user.

---

## Ranking Conflict

In expected-output.txt file, we have following text

1. Tarantulas, 6 pts
2. Lions, 5 pts
3. FC Awesome, 1 pt
4. Snakes, 1 pt
5. Grouches, 0 pts

We have a tie at rank 3, so Grouches should be at rank 4 instead of 5. As two teams are sharing a rank, in our example rank 5 should never exist. As per our logic, Grouches is at 4rth rank.

---

## Test Case Flow

**We have added unit test for all methods call in flow**

1. Before each test, we need to initialize our class.
2. In `testparsegames_input`, we are reading the input file `sample_input_test.txt` and populating a predefined data structure.
3. In `testcalculategames_result`, we are calculating the points and sorting the teams as per calculated points
4. In `testgenerateoutput`, we are calculating the ranks and displaying the final output at console.