



# EAD CA3

**Ryan Deering**  
Lecturer: Gary Clynch

Azure App Service Deploy URL: <http://eadca3-1-dev-as.azurewebsites.net/>

GitHub URL: <https://github.com/ryandeering/EADCA3/>

### Screenshots:









Football Stats

League Tables

Matches Today

Choose a country's domestic league you'd like to see the results for:

English Premier League

Position	Club	Played	Won	Drawn	Lost	GF	GA	GD	Points	Form
1	 Tottenham Hotspur FC	11	7	3	1	23	9	14	24	W,D,W,W,W
2	 Liverpool FC	11	7	3	1	26	17	9	24	W,D,W,D,W
3	 Chelsea FC	12	6	4	1	25	12	13	22	W,D,W,W,W
4	 Leicester City FC	11	7	0	4	21	15	6	21	W,L,L,W,W
5	 West Ham United FC	12	6	2	4	20	15	5	20	W,L,W,W,W
6	 Southampton FC	11	6	2	3	21	17	4	20	W,L,D,W,W
7	 Manchester United FC	11	6	2	3	19	17	2	20	D,W,W,W,W
8	 Manchester City FC	11	5	4	2	17	11	6	19	D,W,W,L,D

Football Stats

League Tables

Matches Today

Choose a country's domestic league you'd like to see the results for:

English Premier League

Input the date for the match results you want to see:

12/12/2020

Time	Status	Home	Score	Away
12:30:00	FINISHED	Wolverhampton Wanderers FC	0 - 1	Aston Villa FC
15:00:00	FINISHED	Newcastle United FC	2 - 1	West Bromwich Albion FC
17:30:00	FINISHED	Manchester United FC	0 - 0	Manchester City FC
20:00:00	IN_PLAY	Everton FC	1 - 0	Chelsea FC

## Description of System:

Application has two pages. The first, allows for a user to check the current standings of any football team in the 'top 5' domestic football leagues in Europe. I wanted to include the League of Ireland, but the API didn't support it. As you can see from the screenshots above, the League Tables page shows data like the club's position, how many games played, Won/Drawn/Lost, and statistics like Goals For (Goals scored by a team), Goals Against (conceded). [I used Sky Sports as a reference](#). This is very useful for any football fan. Images are loaded through SVG thanks to the API. The API is from [football-data.org](https://football-data.org).

The second page shows if there's any matches on today, you can select the different league and date you want. So, if you want to say check the results for the 26<sup>th</sup> of December 2019, where Liverpool beat Leicester 0-4. Or, if you wanted to check a specific date for how Borussia Dortmund, Paris Saint-Germain or Inter Milan did on a certain day, you can as well. The mix and match searching are also useful for any football fan. Unfortunately, there is no SVG images which would have looked nice to include.

## Deploy and Version Control:

The app is deployed on Azure App Service, in a Release configuration. I wanted to have different error messages depending on whether the project was deployed in a Development or Release mode, but it sadly interfered with the Test project. I had a continuous deployment pipeline set up with Azure Devops and Github during development, but it stopped working when I upgraded the project to .NET 5. I am still able to deploy from Visual Studio, and what is being hosted is the latest version of the project. GitHub was used for version control.

## Code Quality/Metrics:

I kept an eye on Solarlint during development of this CA, and I am happy to report there are no Solarlint warnings. I suppressed some relating to the storing of keys and API links, but this does not apply to client-side applications because there is no real secure way to store such secrets in Blazor Webassembly.

I used partial class files for the Razor Pages and put my model classes into the Shared folder to make everything accessible and easy to read. Embedding too much C# code in a Razor Page file is just ugly, in my opinion.

## Testing:

I used xUnit and [bUnit](#), the latter being a community-developed testing library for Blazor components. It's cool and allows for the injection of Blazor services. Unfortunately, it seems still very immature to use. Blazor Webassembly does not have a lot in the way of testing now from what it looks like, as this seemed like the only viable way of creating unit tests. I used bUnit to test the components, Index and MatchesToday. I also used xUnit to test my JSON parsing method, which calls the API.