Tactalyse-2
Testing Document

Sangrok Lee (S3279480)

Bianca Raețchi (S4755138)

Matteo Gennaro Giordano (S5494974)

Mikko Brandon (S3781356)

2023

RUG

TRACEABILITY MATRIX

REQUIREMENT	MODULE	FILES AFFECTED	TEST	PASSED?
		PDF Generator		
US-M1-RQ-F1: Frontend service must be able to pass excel files to the PDF generator.	controller	app.py	test_pdf_endpoint, test_pdf_enpoint_compare	✅
US-M1-RQ-F2: Target player's match data extracted from its file must be contained in a dataframe, which will be used as input data for the graph generator and PDF generator.	data, controller	excel_reader.py, app.py	test_player_data, test_pdf_endpoint, test_pdf_endpoint_compare	✅
US-M2-RQ-F1: Numeric excel data must be turned into line plots, bar plots and radar charts which contain players' data.	All	All	test_pdf_endpoint test_pdf_endpoint_compare	✅
US-M2-RQ-F2: The graphs must be in PNG format that can be included in the PDF.	graph_generator	bar_plot.py, clustered_bar_plot.py, leaderboard_bar_plot.py, line_plot.py	test_image_format, test_image_format, test_image_format, test_draw_returns_png	✅
US-M2-RQ-F3: The graphs must include only relevant stats for the passed player, based on the dictionary of relevant stats for each player position that Tactalyse provided us with.	data	preprocessor.py, excel_reader.py, line_processor.py	test_league_stats_GK, test_league_stats_def, test_league_stats_atk, test_league_stats_ST, test_read_file, test_get_stats_GK, test_get_stats_FB, test_get_stats_CB, test_get_stats_DM, test_get_stats_AM, test_get_stats_WI, test_get_stats_ST	✅

US-M3-RQ-F1: All graphs must contain two players' match data depending on if a comparison PDF was requested.	graph_generator	line_plot.py, bar_plot.py, clusteres_bar_plot.py, leaderboard_bar_plot.py	(tests that check if two dataframes were passed)	✅
US-M3-RQ-NF1: The difference between the data of the two players within the graphs must be clearly visible.	graph_generator		Visual	✅
US-M4-RQ-F1: General player information contained in football league data must be turned into a string format that will be used for displaying basic information of the player(s) at the start of the data report.	pdf_generator	comparison_pdf, standard_pdf	test_print_player_info, test_print_player_info	✅
US-M5-RQ-F1: After employees pass the league and player excel files along with additional parameters, the service must return a complete PDF file to the employees.	All	All	test_pdf_endpoint test_pdf_endpoint_compare	✅
US-M5-RQ-F2: There must be 2 different kinds of PDFs available based on input parameters: non-comparison and comparison reports.	controller	pdf_service	test_create_pdf_no_compare, test_create_pdf_compare	✅

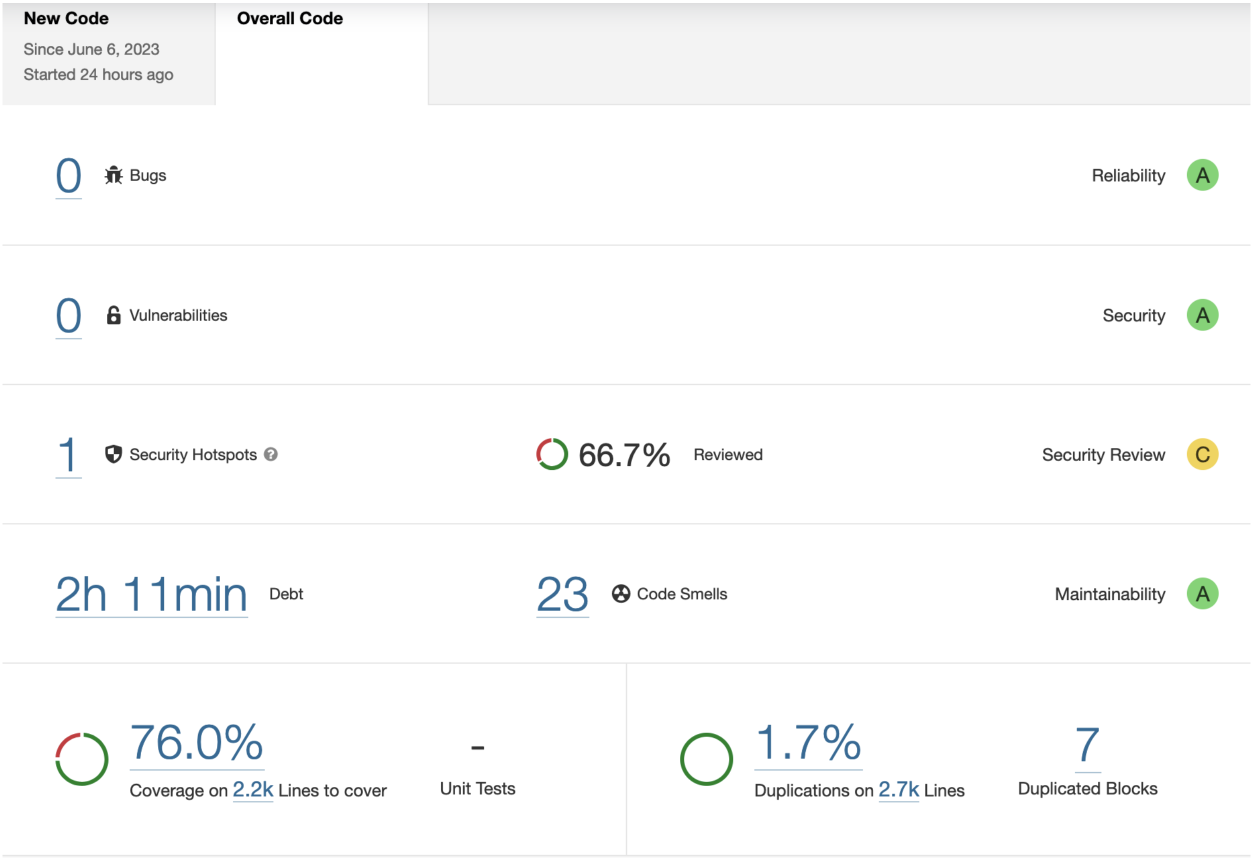
Graph Generator				
US-M6-RQ-F1: If only the graph type ‘radar’ is passed and no additional information, the data for the graph must be randomly selected for two players.	controller, data	app.py, radar_graph_service.py, data_connector.py, randomizer.py, graph_connector.py	test_app.py, test_radar_graph_service.py, test_data_connector.py, test_randomizer.py, test_graph_connector.py	✅
US-M6-RQ-NF1: The graph must include the names of ethe players that were randomly chosen for the data.	controller	app.py, radar_processor.py, line_processor.py	test_app.py, test_radar_processor.py, test_line_processor.py	✅
US-M7-RQ-F1: When the graph type ‘radar’ and target players are passed, the radar chart for the selected players must be generated.	All	app.py, radar_graph_service.py, data_connector.py, randomizer.py, graph_connector.py, radar_chart_factory.py, preprocessor.py, radar_processor.py, radar_chart.py	test_app.py, test_radar_graph_service.py, test_data_connector.py, test_randomizer.py, test_graph_connector.py test_radar_processor.py, test_radar_chart.py	✅
US-M8-RQ-F1: When the graph type ‘line’ and one target player are passed, the line plot for the selected player must be generated.	All	app.py, line_graph_service.py, data_connector.py, randomizer.py, graph_connector.py, line_plot_factory.py, preprocessor.py line_processor.py, line_plot_data_helper.py line_plot.py	test_app.py, test_line_graph_service.py, test_data_connector.py, test_randomizer.py, test_graph_connector.py test_line_processor.py, test_line_plot_data_helper.py test_line_plot.py	✅
US-M8-RQ-F2: When the graph type ‘line’ and two target players are passed, the line plot for the selected players must be generated in the comparison version.	All	app.py, line_graph_service.py, data_connector.py, randomizer.py, graph_connector.py, line_plot_factory.py, preprocessor.py line_processor.py, line_plot_data_helper.py line_plot.py	test_app.py, test_line_graph_service.py, test_data_connector.py, test_randomizer.py, test_graph_connector.py test_line_processor.py, test_line_plot_data_helper.py test_line_plot.py	✅

US-M9-RQ-F1: The graph generator must be able to fully randomize data to generate a plot.	data	randomizer.py, random_graph_service.py, data_connector.py	test_randomizer.py, test_random_graph_service.py, test_data_connector.py	✅
US-M9-RQ-F2: The random graph must either be a radar chart, or a line plot.	controller, graph_generator	graph_factory.py, data_connector.py	test_data_connector.py	✅
US-M9-RQ-F3: When a line plot is randomly generated, it must contain data for only one player.	data, graph_generator	graph_factory.py, randomizer.py, preprocessor.py, line_processor.py	test_randomizer.py, test_line_plot.py, test_line_plot_data_helper.py test_line_processor.py	✅
US-M9-RQ-F4: When a radar chart is randomly generated, it must be a comparison graph containing two players' data.	data, graph_generator	graph_factory.py, randomizer.py, preprocessor.py, radar_processor.py	test_randomizer.py, test_radar_chart.py test_radar_processor.py	✅
Could Have				
US-C1-RQ-NF1: After sending a request to the PDF generator, the PDF should be generated within 20 seconds, as decided by Tactalyse.	PDF generator - All	All	test_pdf_endpoint_speed test_pdf_endpoint_compare_speed	✅

Sonarqube Report

We evaluated our products with Sonarqube, a tool for testing code quality through static code inspection. In this section, we show the results, and address them. We ran the scan iteratively, improving the code based on the tool's feedback.
(Note that unit tests have been implemented, even though the report shows "-" as the number of unit tests. Moreover, the actual percentage of test coverage should be higher since sonarqube apparently has scanned the tests themselves in search for coverage.)

PDF GENERATOR



New Code
Since June 6, 2023
Started 24 hours ago

Overall Code

0 Bugs — Reliability A

0 Vulnerabilities — Security A

1 Security Hotspots — 66.7% Reviewed — Security Review C

2h 11min Debt — 23 Code Smells — Maintainability A

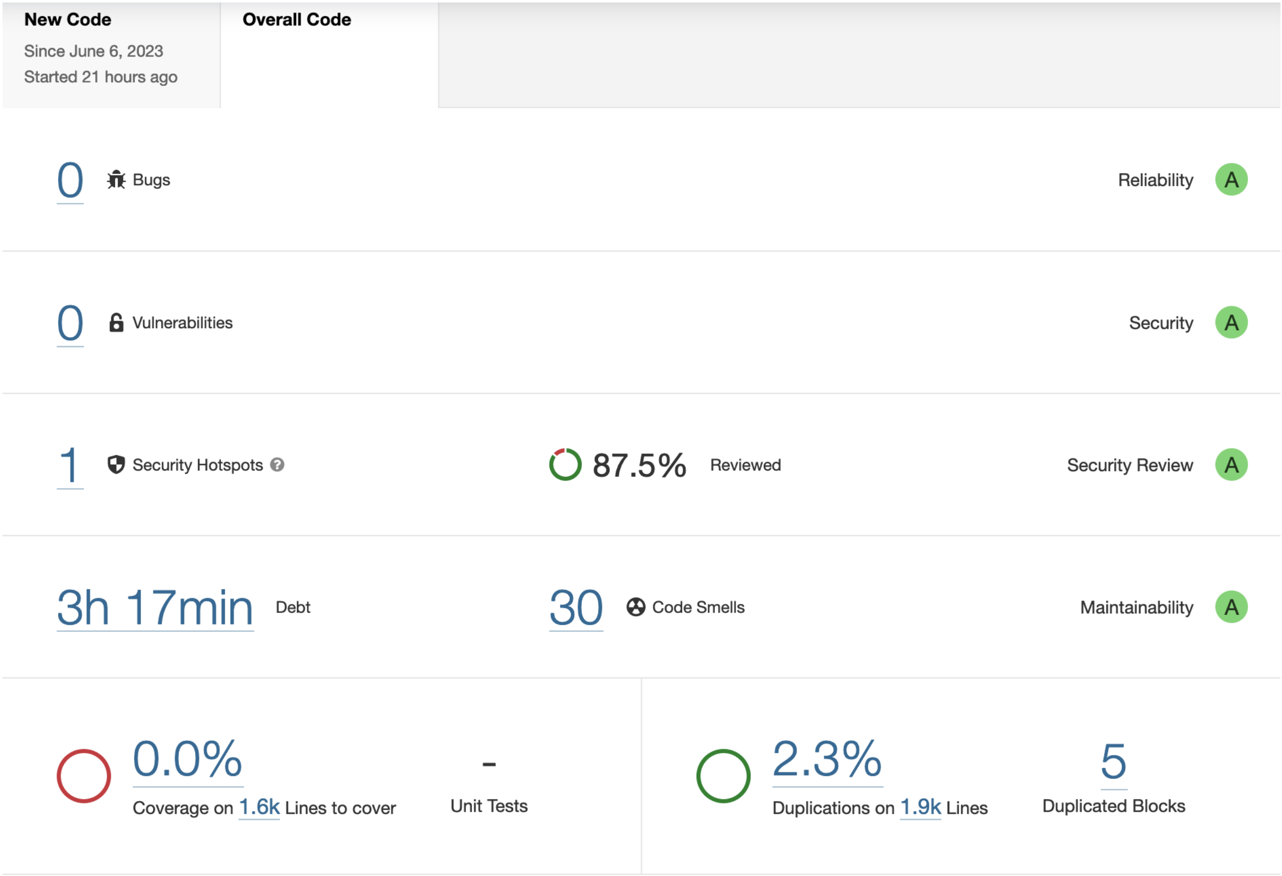
76.0% Coverage on 2.2k Lines to cover — - Unit Tests

1.7% Duplications on 2.7k Lines — 7 Duplicated Blocks

Figure 1: Sonarqube report of the PDF generator API code

The remaining code smells are mostly due to the duplication of strings representing player statistics in the unit tests.

GRAPH GENERATOR



New Code
Since June 6, 2023
Started 21 hours ago

Overall Code

0 Bugs — Reliability A

0 Vulnerabilities — Security A

1 Security Hotspots — 87.5% Reviewed — Security Review A

3h 17min Debt — 30 Code Smells — Maintainability A

0.0% Coverage on 1.6k Lines to cover — - Unit Tests

2.3% Duplications on 1.9k Lines — 5 Duplicated Blocks

Figure 2: Sonarqube report of the graph generator API code

(Unit tests have been implemented, as documented in the traceability matrix, even though the report shows 0% test coverage)

It should be noted that the security hotspots in the reports are concerns about CSRF protection being disabled. Such protection is still disabled and not implemented in the final submission of our codebase, as detailed in the Will Not Have user stories in the requirements section.

Acceptance testing

Our apps are purely backend APIs. This means that we did not develop a user interface for the client to interact with. As such, our acceptance criteria are purely visual, and related to the design of the PDF reports and the graphs, i.e. the output of the APIs.

We tried to have the client fill out the below form to account for acceptance testing. They communicated that they would do it, but by the time of submission they had not gotten around to it, despite multiple attempts at reminding them. As such, the form itself is empty. We will send an updated version once/if it is filled in by the client.

<u>Explanation</u>
Criterion: the visual element we had the client judge.
Feedback: the received feedback for the specified visual element, if any.
Unsatisfied requirements: requirements the client communicated that were missing from the final product, if any.
Passed?: yes/no indicating whether it satisfied the client's expectations.
Other remarks: field for any additional comments unrelated to the previous fields.

All final versions of graphs and reports are included in the appendix, section B.

Line plot

1.

Criterion	Tactalyse logo placement
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

2.

Criterion	Title and subtitle
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

3.

Criterion	Colors and design of all lines
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

4.

Criterion	x-axis (seasons) and y-axis (stat values)
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

5.

Criterion	Legend
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

6.

Criterion	General readability of the (comparison) graph
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

Radar chart

7.

Criterion	Tactalyse logo placement

Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

8.

Criterion	Title and subtitle
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

9.

Criterion	Grid lines (the gray circles and lines)
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

10.

Criterion	Stat labels
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

11.

Criterion	Player plot lines
Feedback	
Unsatisfied requirements	
Passed?	

Other remarks	

12.

Criterion	Legend
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

13.

Criterion	General readability of the (comparison) graph
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

Single Player PDF Report

14.

Criterion	Logos at the top of the page
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

15.

Criterion	Front page title
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

16.

Criterion	Player information and image
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

17.

Criterion	Page design (background, colors etc)
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

18.

Criterion	Line plot section title and subtitle
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

19.

Criterion	Plot placement on the pages
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

20.

Criterion	Line plot title and subtitle
Feedback	

Unsatisfied requirements	
Passed?	
Other remarks	

21.

Criterion	General line plot readability in the report
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

22.

Criterion	Bar plot section title and subtitle
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

23.

Criterion	Bar plot colors
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

24.

Criterion	Bar plot colors
Feedback	
Unsatisfied requirements	

Passed?	
Other remarks	

25.

Criterion	Bar plot legend (gradient scale on the left)
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

26.

Criterion	Ranking bar plot: design, bar shape and orientation
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

27.

Criterion	Ranking bar plot: x-axis (stat values) and y-axis (player names)
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

28.

Criterion	General readability of ranking bar plots
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

29.

Criterion	Standard bar plot: design, bar shape and orientation
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

30.

Criterion	Standard bar plot: x-axis (player names) and y-axis (stat values)
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

31.

Criterion	General readability of standard bar plots
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

31.

Criterion	Summary bar plot: design, bar shape and orientation
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

31.

Criterion	Summary bar plot: x-axis (player names) and y-axis (stat values)
Feedback	

Unsatisfied requirements	
Passed?	
Other remarks	

31.

Criterion	General readability of summary bar plots
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

Comparison PDF Report

32.

Criterion	Front page title
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

33.

Criterion	Player information and images
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

34.

Criterion	General line plot readability in the report
Feedback	
Unsatisfied requirements	

Passed?	
Other remarks	

35.

Criterion	General readability of ranking bar plots
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

36.

Criterion	General readability of standard bar plots
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

37.

Criterion	General readability of summary bar plots
Feedback	
Unsatisfied requirements	
Passed?	
Other remarks	

Appendix

A. Change Log

- 02 June 2023:
    - **Matteo**: Created traceability matrix
    - **Mikko**: Filled traceability matrix for PDF generator
- 05 June 2023:
    - **Bianca**: Outlined relevant use cases for graph API tests in traceability matrix
- 06 June 2023:
    - **Matteo**: Worked on Sonarqube Report section
    - **Mikko**: Worked on Acceptance Testing section
- 07 June 2023:
    - **Mikko**: Revised Acceptance Testing section with feedback from Andrea, moved final product versions to appendix
    - **Matteo**: Changed Sonarqube Report section to only include the final reports, filled traceability matrix for graph generator
- 9 June 2023: **Bianca, Sangrok, Mikko, Matteo**: Finished the document

B. Final Product Versions

This section contains examples of the final output of the PDF generator and graph generator APIs.

Reports:

- Single player PDF report: single_final.pdf
- Comparison PDF report: compare_final.pdf

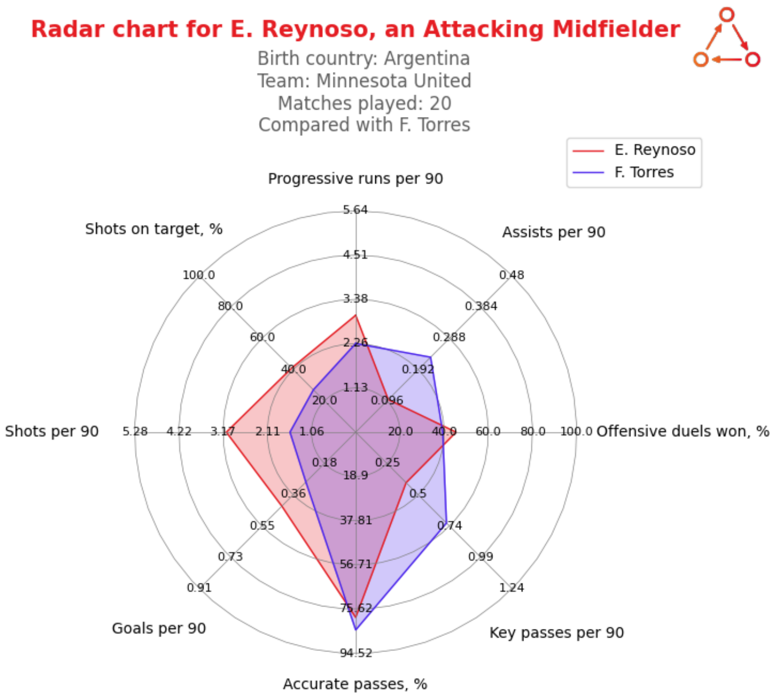


Figure B1: Final version of a comparison radar graph, as output by the graph generator

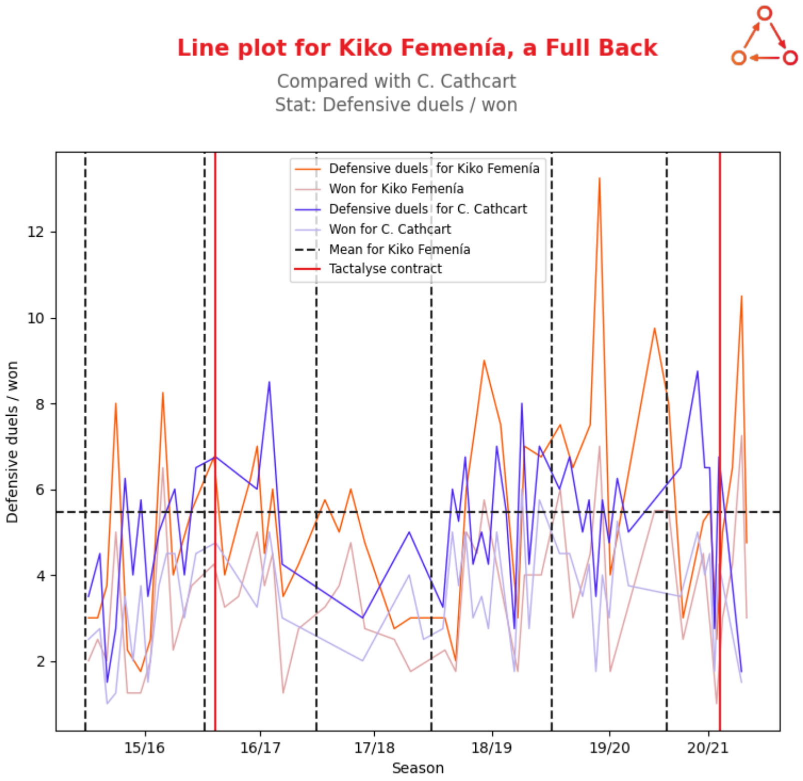


Figure B2: Final version of a comparison line graph, as output by the graph generator

C. Conceptual Prototypes

After we set up a code skeleton and basic functionality for the project, we started working on graph prototypes to explore ways of visualizing football data. This section showcases the prototypes that were created in this phase.

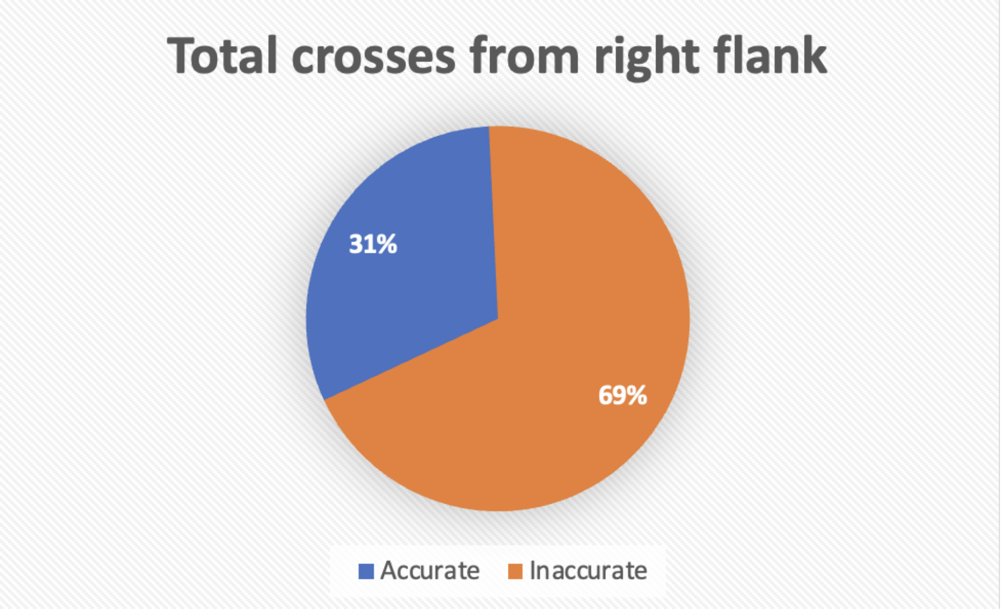


Figure C1: Pie chart showcasing a league stat for one player, with the slices indicating a proportion of the substat vs the total. Scrapped in conceptual phase.

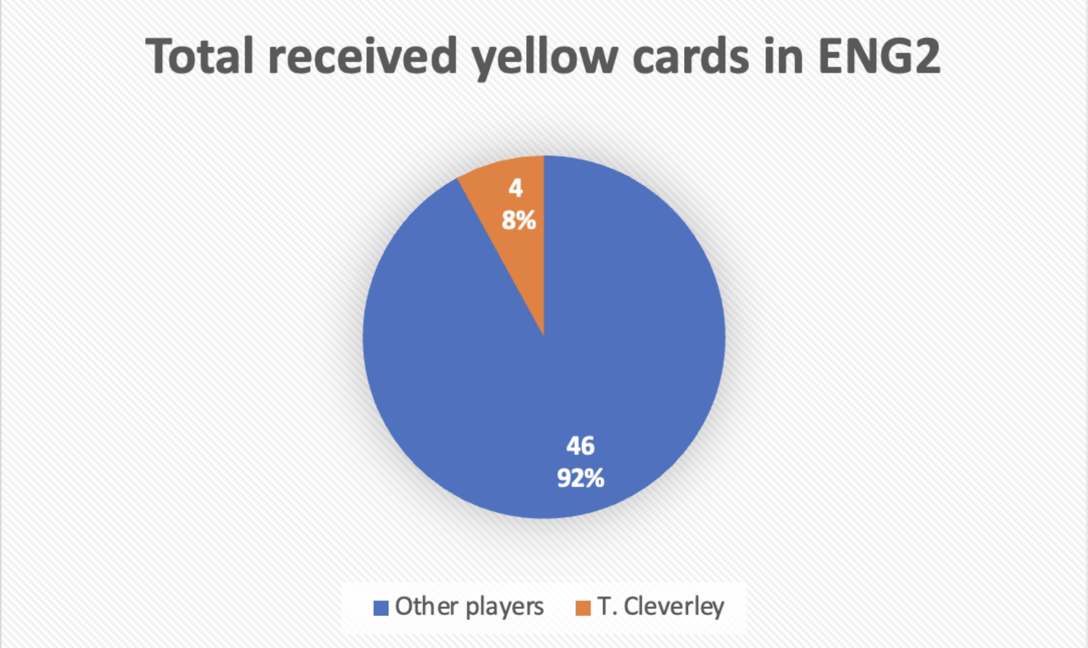


Figure C2: Pie chart showing the percentage of a stat total in a league that a player’s total amount makes up. Scrapped in conceptual phase.

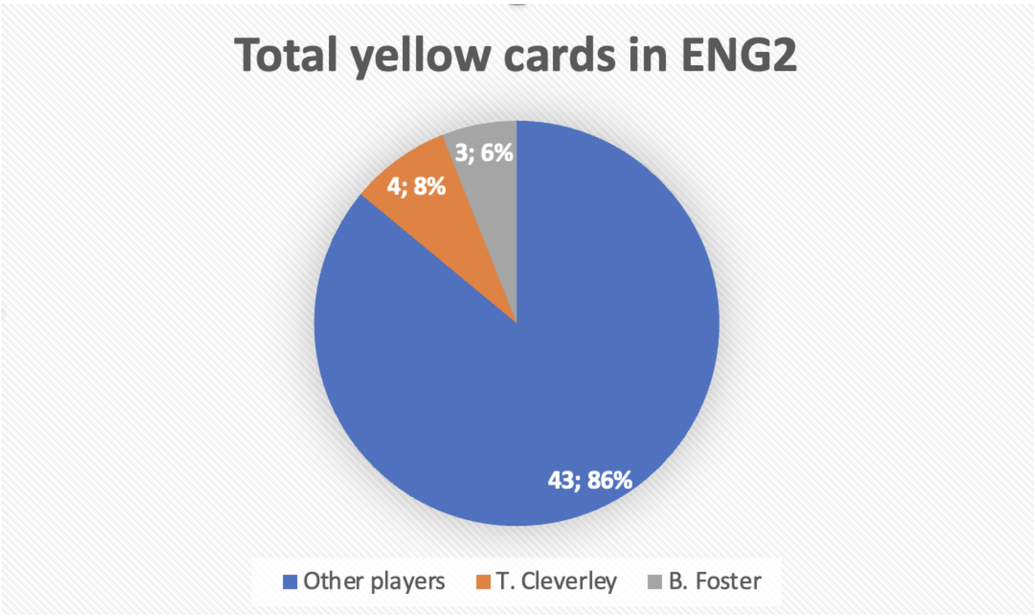


Figure C3: Pie chart showing the percentage of a stat total in a league that a player's total amount makes up, with comparison. Scrapped in conceptual phase.

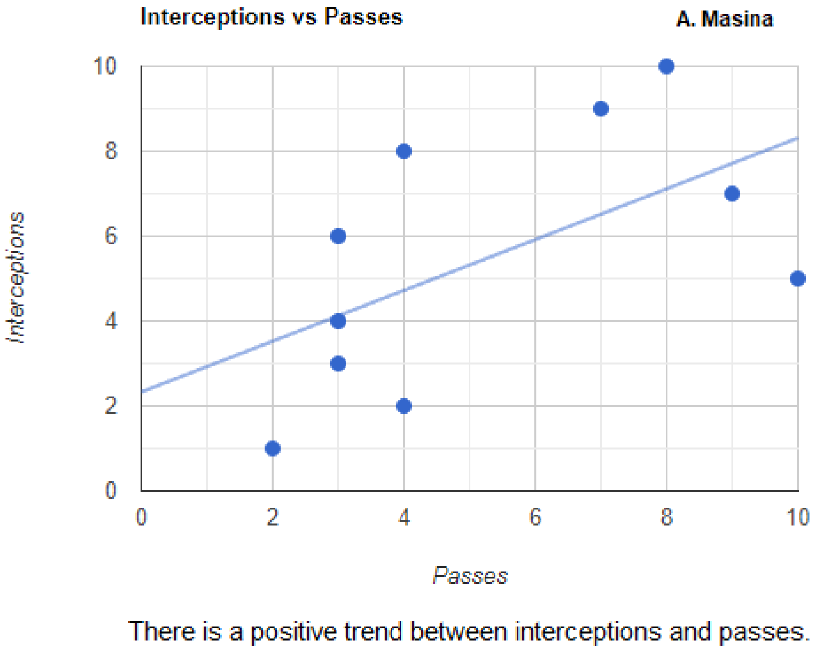


Figure C4: Scrapped correlation plot. Implemented in code but not included in the final PDF report versions.

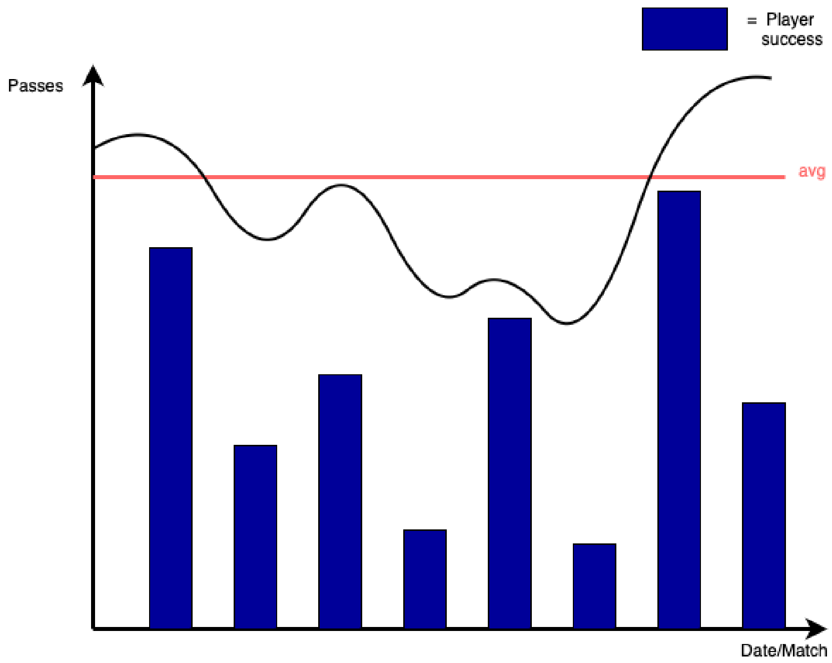


Figure C5: Prototype of a single player line/bar combination plot. Partially approved by the client, and used for implementation of the final line graph.

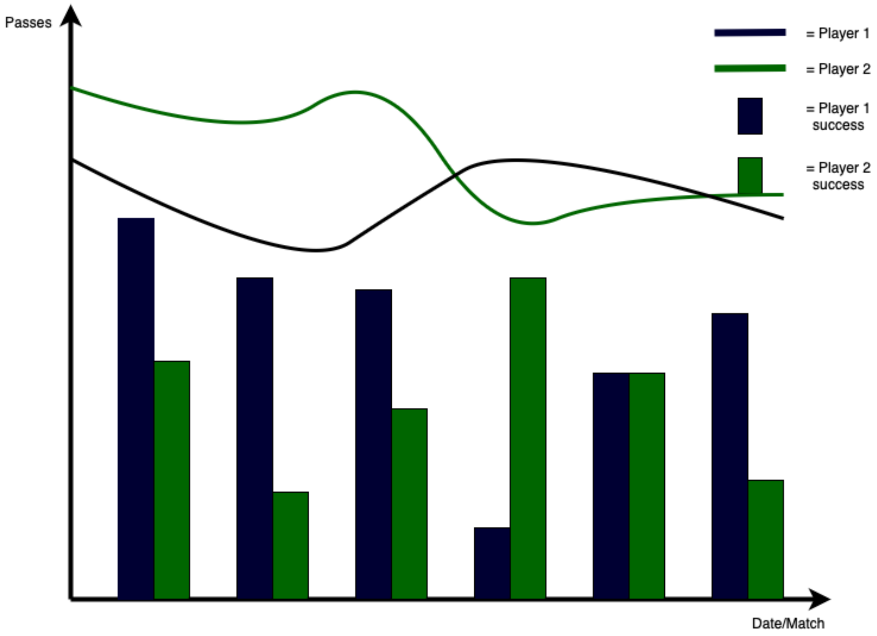


Figure C6: Prototype of a comparison line/bar combination plot. Partially approved by the client, and used for implementation of the final line graph.

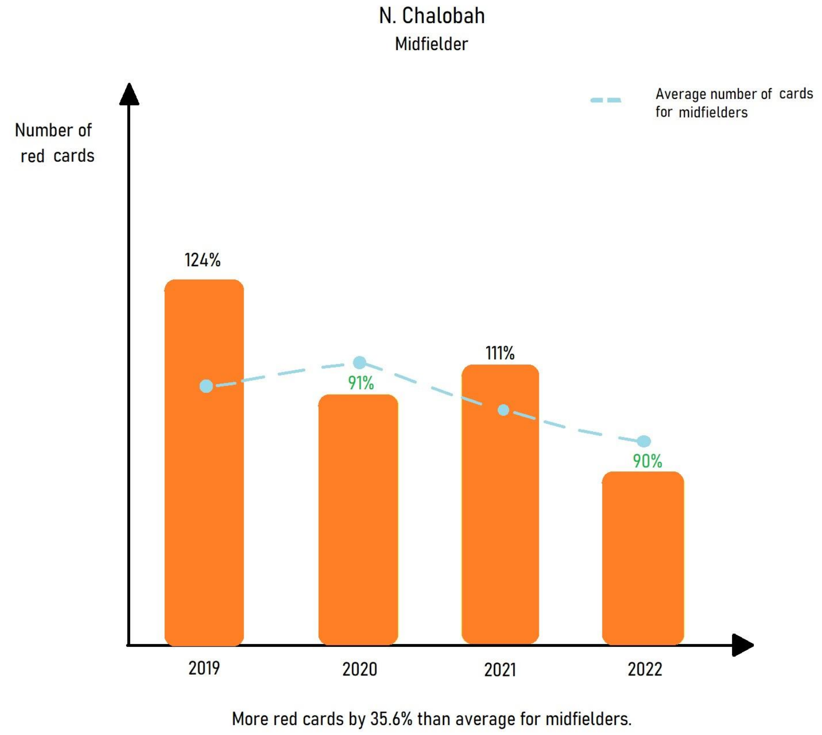


Figure C7: Prototype of a temporal single player bar plot. Partially approved by the client, and used for implementation of the final bar plot.

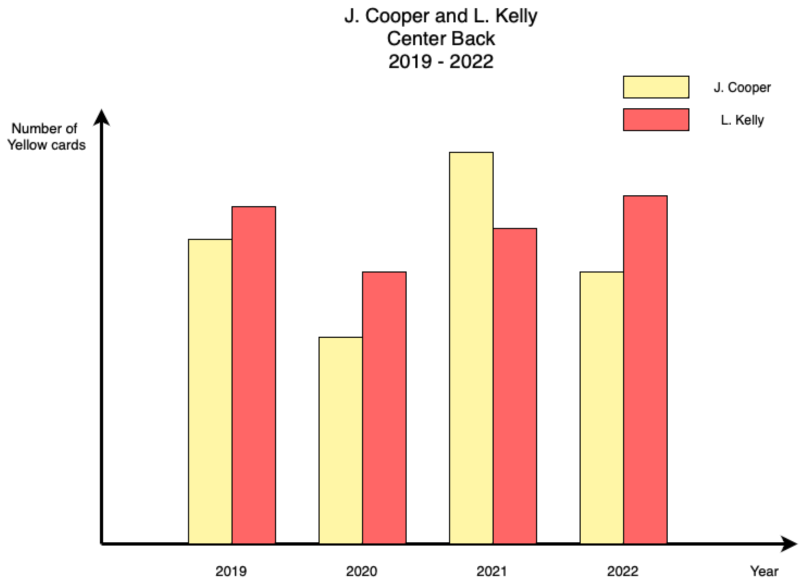


Figure C8: Prototype of a temporal comparison bar plot. Partially approved by the client, and used for implementation of the final bar plot.

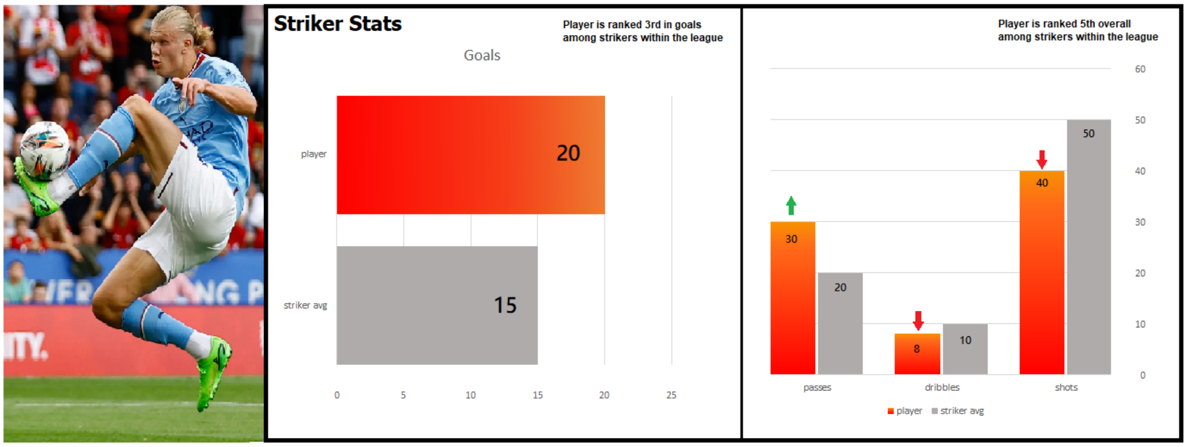


Figure C9: Prototype of a collection of bar graphs showcasing a player's positional statistics and rankings in a league. Partially approved by the client, and used for implementation of the final ranking bar plot.