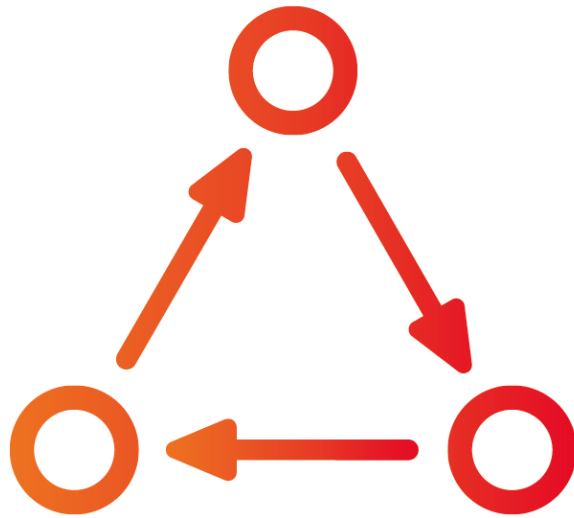# Tactalyse-2 Requirements Document

Sangrok Lee (S3279480)

Bianca Raețchi (S4755138)

Matteo Gennaro Giordano (S5494974)

Mikko Brandon (S3781356)

2023

RUG

Contents

Glossary

Term	Definition
Football	The european term, not to be confused with American football.
Player position	Indicates football positions such as goalkeeper, winger, midfielder, etc.
(The) service	An abstraction of the PDF Generator application
Graphs/Plots	Radio charts and line plots containing player statistics.
Radio chart	A circular plot that showcases parameter values on axes of the circle.
Coaches	Football coaches of professional football clubs.
Players	The football players the coaches request Tactalyse's service for.
(The) client	Football coaches, Tactalyse's clients.

1. Introduction

Tactalyse is an organization that aims to provide data reports about football players. The data report contains statistics and graphs based on the football players' match data. To build those graphs and statistics, coaches who are customers of Tactalyse upload the players' data in excel file format on the company's website. Then, employees of Tactalyse receive the data and upload it to a separate page on the website, accessed only by them. The website generates a data report. This generated data report will be e-mailed to coaches for improving the players' performance. In this project, our main task is to receive data from the frontend service, create related graphs and plots based on the received data, and create data reports from these graphs.

The company already has a completed mechanism that receives data and creates related graphs and reports. However, the original system has several bugs, and is written with bad code logic and with no architecture in mind. Additionally, the used libraries are not optimized, and very slow. Since data report generation speed has a great impact on the company's work performance, the owner wants us to build their project from scratch, rather than extend the original project.

2. Stakeholders

For this project, there are multiple stakeholders who will be affected by our app. In order to make sure all of their requirements are satisfied, a description of their potential expectations is provided.

Actors

- **Tactalyse employees**: they will be the main users of our app and they will send the clients the generated reports. Therefore, creating an *intuitive* and *efficient* app for the employees to work with is going to reduce the potential for errors in the future and should be kept at the forefront of our design choices.

General Stakeholders

- **Owner of Tactalyse**: this stakeholder owns the app. In order to keep them satisfied, *deadlines should be followed*, and their *desired features should be included* in the end product. The owner particularly values their time, so the app needs to be *fast* as well.
- **Coaches**: they will receive the reports generated by the employees. Since they are the main clients of Tactalyse, it is crucial to make sure this stakeholder's requirements are at the top of our priority list. It can be expected that they will favor data reports with a *clean layout*, so the information displayed will be easy and straightforward to understand.

- **Players**: the players are indirect stakeholders of our application. They will be impacted most by their coaches' interpretation of our data reports. In order to have a better understanding of their current athletic abilities, providing a *comprehensive selection* of such reports would be the most beneficial component.

3. Personas

For this project, the persona we're focusing on is Tactalyse's employees, as they are the only group interacting with our app. Included below is a breakdown of how the employees are supposed to use the final product.

- **The employee**: a person that is granted access to the app by Tactalyse and whose job is to deliver data reports for Tactalyse's clients. The employees have access to the API serving our app through an existing interface, which contains a form that has to be filled in, along with file inputs for providing datasheets. By providing this information and submitting it through the interface, the existing app will call the necessary endpoint and redirect it to the app, where a report will be generated based on that information and return the resulting PDF. The employee will not be able to edit the report, or its general layout. The only possible type of interaction will be the generation of the data report.

4. Epics

This section describes the epics associated with the project. An epic consists of a set of features that together add up to a service.

Epic 1: Data Report PDF Generator

Tactalyse's owner wants to provide a fast service that delivers PDF reports of football data. An employee of Tactalyse uploads data files to our backend. The backend extracts data from the files, generates graphs based on the extracted data, and generates a single PDF report including the generated graphs and extracted data. This PDF report is returned to the employee.

5. Features

This section describes the features that can be extracted from the epics. Features can be seen as a higher-level grouping of user stories. As such, user stories are listed under the relevant feature.

- Convert excel file to usable data
    - **US-M1**
- Create statistical analysis out of input data
    - **US-M2, US-M3, US-M4**
- Generating multiple types of pdf documents with different statistics for different purposes
    - **US-M2, US-M5**
- Overall improved PDF generation service
    - **US-S1**

6. User Stories

This section details user stories. The user stories are grouped using the MoSCoW scheme. Each user story is accompanied by associated requirements, as well as acceptance criteria for that specific user story.

Must have

US-M1: As an employee of Tactalyse, I would like to be able to extract data from excel files so that I can turn the data into related graphs or plots.

Requirements:
- **US-M1-RQ-F1** Frontend service should be able to pass excel files to our service.
- **US-M1-RQ-F2** Data should be extracted from the passed excel files.
- **US-M1-RQ-F3** It should be possible to extract data from any excel file, regardless of the type of data (football or not).
- **US-M1-RQ-F4** Extracted data should be contained in a dataframe which will be used as input data for the graph generator and pdf generator.

Acceptance criteria:
- **Given:** the backend controller has been provided with an excel file.
- **When:** the controller passes the file to the data processor.
- **Then:** the data from the excel file should be available in dataframe.

US-M2: As an employee of Tactalyse, I would like to be able to generate graphs and plots so that I can communicate data to clients.

Requirements:
- **US-M2-RQ-F1** Numeric excel data should be turned into radio charts and line graphs which contain player's data.
- **US-M2-RQ-F2** The graphs should be in a data format that can be included in the PDF.
- **US-M2-RQ-F3** The graphs should be appropriate for the passed player position.
- **US-M2-RQ-NF1** The graphs should adhere to best practices when it comes to their design.

Acceptance criteria:
- **Given:** the service has extracted players' data.
- **When:** the graph generator receives the players' data.
- **Then:** the graph generator should output the desired graphs and plots.

US-M3: As an employee of Tactalyse, I would like to have an overview of two players' data in the same graphs so that I can compare two football players.

Requirements:
- **US-M3-RQ-F1** All graphs should contain two players' data depending if a comparison PDF was requested.
- **US-M3-RQ-NF1** The difference between the data of the two players within the graphs should be visible to football coaches.

Acceptance criteria:
- **Given:** the service has been provided with a second player's data.
- **When:** the graph generator receives two players' data.
- **Then:** two players' data should be represented in the generated plots.

US-M4: As a football coach, I would like to have a basic text overview of relevant player stats so that I can identify the correct player in the data report.

Requirements:
- **US-M4-RQ-F1** Player data should be turned into a text format that will be used for displaying basic information of the player at the start of the data report.
- **US-M4-RQ-F2** The text should be relevant to the player's position.

Acceptance criteria
- **Given:** the service has been provided with players' data.
- **When:** the service passes the data to the pdf generator.
- **Then:** the generated pdf should include some of the data in text form in an introductory page.

US-M5: As an employee of Tactalyse, I would like to be able to generate a PDF out of created graphs and text so that I can provide the product to clients.

Requirements:
- **US-M5-RQ-F1** After passing league and player excel files and the player's name, the service should return a complete PDF file.
- **US-M5-RQ-F2** There should be 3 different kinds of PDFs available based on input parameters: Tactalyse client, non-Tactalyse client, and comparison reports.

Acceptance criteria:
- **Given:** graphs have been generated based on input data.
- **When:** the pdf generator receives the graphs and data.
- **Then:** the service should output the final pdf including the generated graphs and data.

Should have

US-S1: As the owner of Tactalyse, I want to have a fast-working PDF service so that I can deliver my product in a timely manner.

Requirements:
- **US-S1-RQ-F1** After sending a request to the service, the PDF should be generated within 5 seconds.

Acceptance criteria:
- **Given:** the Tactalyse employee has been provided with an excel file and additional information that needs to be turned into a PDF format.
- **When:** a PDF is requested on the website.
- **Then:** the PDF should be returned within 5 seconds.

Could have

As of the delivery of this document, no "could have" user stories have been identified.

Will not have

As of the delivery of this document, no "will not have" user stories have been identified.

7. System Use Cases

Use cases are used to formulate one or more grouped requirements in more detail. In this section, use cases of the project are depicted by describing the expected scenario. Each use case includes a context section that tells the basics of the use case, an evaluation of how much the feature will be used, and preconditions and postconditions for the feature. If there is a case of failure for the scenario, the exception scenario will be described.

Use Case - 1 : Convert Excel File to Usable Data

User Story

US - M1

Context

Given the requirements, this is the most basic functionality. Since the excel file contains raw data, the data needs to be converted into a format that's usable for graph generation.

Frequency of occurrence

High. Our service will be started with this use case.

Precondition

Excel file has been transferred from Frontend service.

Postcondition

Excel file is transformed into the usable data format.

Main Success Scenario

1. The user inputs an Excel file into the web service (frontend)
2. The backend receives the Excel file and transforms the file into a usable data format

Exception Scenario

1. The user inputs the wrong file (not an Excel file)
2. Errors during reading of the input file

Use Case - 2 : Generate Graphs/Plots

User Story

US - M2, US - M3, US - M4

Context

Given the requirements, this is basic functionality. The application should be able to make graphs/statistical plots visualizing the player's statistics. These graphs and plots will be used for generating pdfs.

Frequency of occurrence

High. The main purpose of our service is providing data reports based on the player data. To achieve this, this use case always needs to be performed.

Precondition

The graph service receives correctly formatted input data.

Postcondition

Graphs are generated based on the input data.

Main Success Scenario

1. Raw data transformed into usable data format to generate graphs/plots.
2. Graphs and plots about players' performance are generated based on the extracted data.

Exception Scenario

1. The data does not contain an expected feature (e.g. goals stopped for a goalkeeper).
2. The data used in graphs is not properly formatted.

Use Case - 3: Generate Data Reports

User Story

US - M2, US - M5

Context

Given the requirements, this is basic functionality. The service should be able to build a data report for the players' statistics. The report contains graphs and plots generated in Use Case - 2. There will be several types of reports based on the different purposes.

Frequency of occurrence

High. This use case is the main purpose of our application.

Precondition

Generated graphs and extracted data are sent to the pdf generator.

Postcondition

Visualization of data report with generated statistical analysis

Main Success Scenario

1. Graphs and plots are successfully built

2. Data reports are successfully generated by using graphs and data in a readable layout.

Exception Scenario

1. The data does not contain an expected feature
2. Passed graph images are not in the correct format

Appendix

A   Client Meetings

A.1   17 February Fri 2023 First client meeting with Loran, Ayush

- Attendees: Sangrok, Mikko, Matteo, Bianca
- General client meeting : weekly, Monday, 3:30 - 4:00 pm

1. What is our project? What should our team do for the project?

→ Client has data(excel file) as input and our program should generate a pdf report.

→ our team will basically work for the template of the pdf file, so make it more readable.

→ our team will not work for website tasks, there is another team to do that project.

→ There are some existing reports and code but needs to be rebuilt from scratch.

→ Data report needs to contain all the information needed.

→ Reports should be as simple as possible, and one template for all reports.

2. Framework freedom

→ It depends on our team, if we think another tool is easier and useful, we can use them. However, we need to explain why we chose another framework to work because the client wants to know for future work.

3. Put Deadlines in the team channel so that the client knows the schedule of our project.
4. Send Microsoft team mails of our team mail address to communicate with them.

A.2   20 February Mon 2023 3:30 - 4:00 pm Client meeting with Ayush

- Attendees: Sangrok, Mikko, Matteo, Bianca

Q. Do we have freedom in designing the layout of the pdf

→ Yes, use our imagination

Q. Who are the users?

→ Tactalyse employees create pdf and send it to coaches(so maybe employees and coaches)

Q. What is our task on top of the website

→ nothing yet, but as having consult with professor, we could take another team's task if we want

Q. Ask for clarification on different type of pdf

→ individual, comparison, we can imagine more

Q. What does "making code better" mean?

→ Time reduce for generating pdf, there are some unnecessary functions(code refactor)

A.3   7 March Tue 2023 3:30 - 4:00 pm Client meeting with Ayush, Loran

- Attendees: Sangrok, Mikko, Matteo

Q. Twitter bot

→ Client wants us to focus on our main requirements(pdf generation) and then after this is done, we can work on extra features.

→ Access will be in shared dropbox

Q. Where is the excel file(input data) from?

→ Excel file is from the website(https://wyscout.com/): coaches upload the excel file for the pdf.

Request from the client.

→ There are two websites for the project, and the clients want them in one.

   a.   but this is not gonna be our work, this will be another team's work.

A.4   13 March Mon 2023 3:00 - 3:30 pm Client meeting with Ayush, Loran and Tactalyse-1

- Attendees: Sangrok, Mikko, Matteo, Bianca

Only notes related to our project.

   1. Twitter bot
        a.   Graph generation for Twitter bot will be our responsibility

2. Among the plots in the example pdf, we should focus most on the line plot

3. Speeding up the pdf generation is also an important requirement, current source code generates simple pdf(3-4 pages) in 3-5 minutes which is too slow.

A.5   21 March Mon 2023 3:30 - 4:00 pm Client meeting with Ayush, Loran

- Attendees: Sangrok, Mikko, Matteo, Bianca

Q. Do we need a database for saving excel files from the past?

→ Since the column and features of the data will be the same, therefore it will be a good option to implement a database for the future. However, it depends on us.

Q. During the meeting with our TA, we discussed having one API for two endpoints since the pdf generation function contains graph generation. Is it fine to have one API for two of those endpoints?

→ It is fine to have one API for two endpoints. We explained our choice for this and the client agreed with our decision.

A.6   27 March Mon 2023 3:30 - 4:00 pm Client meeting with Ayush, Loran

- Attendees: Sangrok, Matteo, Bianca

Q. There are some players who have more than one position in the excel files. In this case, based on which position should the data report for that player be made?

→ Almost all players have their main position and sometimes play for other positions. We can choose the leftmost position in the excel files as the main position and make a data report based on this position.

Q. There are statistics in the excel files, do we need to follow the format of the excel file in the data report?

→ yes.

Q. Can we add some more basic information for the target player which are not in the example data report?

→ of course yes.

A.7   3 April Mon 2023 4:00 - 4:30 pm Client meeting with Ayush, Loran and Tactalyse -1

- Attendees: Sangrok, Matteo, Bianca, Mikko

- make it visually pleasing

- next meeting: make project plan

- no need to concern Agent this will not be used in further

- Front page(First player basic info page) can be in better layout

B   Changelog

- 27 February 2023 **Sangrok** : Added first, second client meeting.
- 28 February 2023 **Sangrok, Mikko**: First draft of Epics.
- 28 February 2023 **Sangrok, Mikko**: First draft Stakeholders.
- 28 February 2023 **Sangrok, Mikko**: First draft Features.
- 28 February 2023 **Sangrok, Mikko**: First draft User stories based on MoSCoW with requirements(draft needs more opinions from other team members).
- 3 March 2023 **Sangrok, Mikko, Matteo**: Enhance the overall parts in requirements doc(Epics, Features, User Stories)
- 4th of March 2023 **Bianca**: Only persona identified - employee
- 7 March 2023 **Sangrok, Mikko, Matteo**: Add acceptance criteria for all user stories except **US-S2(US-S2** is now deleted because it is too obvious and basic)**.**
- 7 March 2023 **Sangrok**: Add third client meeting.
- 11 March 2023 **Bianca, Sangrok, Mikko, Matteo**: Listed framework choices and added notes about architecture.
- 13 March 2023 **Sangrok**: Add 4th client meeting.
- 22 March 2023 **Sangrok**: Add 5th client meeting.
- 24 March 2023 **Sangrok**: Add Use cases section.
- 28 March 2023 **Sangrok**: Add 6th client meeting, table of content.
- 29 March 2023 **Sangrok:** Modify Introduction part based on the feedback.
- 30 March 2023 **Sangrok**: Add Glossary table.
- 30 March 2023 **Sangrok:** Fixed most parts of the document based on the feedback. (User stories, Use cases)
- 30 March 2023 **Mikko:** Went over/updated Sangrok's changes, added title page.
- 3 April 2023 **Sangrok**: Add 7th client meeting.
- 3 April 2023 **Mikko:** Final fixes.