



UNIVERSIDADE D
COIMBRA

José Pedro Moreira Esperança

FOOTBALL MATCH ANALYSIS PORTAL

Dissertation in the context of the Master in Informatics Engineering, specialization in Software Engineering, advised by Professor Nuno Manuel dos Santos Antunes, PhD, Professor Pedro Manuel Henriques da Cunha Abreu, PhD, and Engineer Arnaldo José da Silva Moura, BSc, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July of 2023



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

José Pedro Moreira Esperança

Football Match Analysis Portal

Dissertation in the context of the Master in Informatics Engineering, specialization in Software Engineering, advised by Professor Nuno Manuel dos Santos Antunes, PhD, Professor Pedro Manuel Henriques da Cunha Abreu, PhD, and Engineer Arnaldo José da Silva Moura, BSc, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July 2023

Acknowledgements

Firstly I would like to thank my advisors: Arnaldo Moura, Nuno Antunes, and Pedro Abreu, for their invaluable support, insightful suggestions, and guidance throughout the duration of this internship.

Secondly, I would like to show my appreciation towards Susana Lourenço and the entire Grama team for their warm welcome and support. I would like to extend a special thanks my fellow interns at Grama, Gonçalo São Marcos and Francisco Bugalho, who despite not being directly involved in this particular internship, were always ready to engage in insightful discussions, offer valuable suggestions, and provide assistance whenever I required it.

Lastly, even though words may not be enough, I would like to thank my family and friends, with an even more special thanks to my mom, my dad, and my sister. You have all shaped me to be the person I am today, given me your unconditional support and, above all, consistently gifted me my fondest and happiest moments.

Abstract

Football boasts a diverse and dedicated fan base, from the more casual spectators to the hardcore ones. The significant majority of them like to keep up with their favorite sport by consuming content about it. Although some might prefer a lighter approach, just keeping up with the scores of the latest matches, others want to delve deep and to thoroughly read reviews and tactical analyses created by their preferred critics.

Currently, however, match reviews and opinions are scattered throughout the entire internet. Whether you are a casual watcher or a professional scout from one of the world's top clubs, finding a review about a particular football match you are interested in becomes quite challenging. This is due to the content spread caused by authors not writing on the same websites. Subsequently, it becomes difficult to filter interesting content pieces from lower quality ones, as it is difficult to compare content dispersed among different platforms.

This internship, which has the duration of a year, focuses on the creation of Golden Goal. This is a new online platform that will feature various content pieces, such as reviews and live experiences from football matches. It aims to solve the previously mentioned problems by centralizing content onto the platform, allowing users to interact with it by rating matches and reviews, and marking matches as watched. Users will also be able to create reviews, experiences, or collections of their favorite content to share with other users of the platform.

Keywords

Football, Reviews, Football Matches, Microservices, Amazon Web Services, Java Spring, React

Resumo

O futebol apresenta uma grande variedade de adeptos, desde os mais casuais aos mais dedicados. A grande maioria destes desfruta de acompanhar o seu desporto favorito e de consumir conteúdo sobre o mesmo. Embora alguns prefiram fazê-lo de uma maneira mais descontraída, apenas vendo os resultados dos jogos mais recentes e das suas equipas favoritas, outros querem aprofundar os temas e ler atentamente as análises táticas criadas pelos seus críticos preferidos.

No entanto, as análises e críticas de jogos estão dispersas por toda a internet. Quer os espectadores casuais quer os olheiros profissionais dos melhores clubes têm dificuldades em encontrar críticas sobre um determinado jogo em que estejam interessados. Isto deve-se aos autores não escreverem todos na mesma plataforma o que, subsequentemente, torna difícil distinguir a qualidade de conteúdos que estão dispersos por várias plataformas diferentes.

Este estágio, com duração de um ano, tem como foco a criação do Golden Goal. Esta é uma nova plataforma *online* que apresenta conteúdo variado como análises e experiências de jogos que os utilizadores tenham visto ao vivo. Esta plataforma visa resolver os problemas mencionados anteriormente através da centralização do conteúdo na mesma, com o qual os utilizadores podem interagir através de avaliações de jogos e análises. Para além disto, os utilizadores do Golden Goal podem criar as suas próprias análises, experiências, ou coleções dos seus conteúdos favoritos para partilhar entre si.

Palavras-Chave

Futebol, Análises, Jogos de Futebol, Microserviços, Amazon Web Services, Java Spring, React

Contents

1	Introduction	1
1.1	Gramma and Golden Goal	1
1.2	Context and Motivation	2
1.3	Objectives	2
1.4	Structure	3
2	State of the Art	5
2.1	Competition Analysis	5
2.1.1	Competitors	5
2.1.2	List of Analyzed Features	6
2.1.3	Results and Discussion	8
2.2	Rating Systems	10
2.2.1	Analyzing a Rating System	10
2.2.2	Comparing rating systems	11
2.2.3	Review Bombing	13
2.2.4	Analysis	14
2.3	Monolithic vs Microservices Architectures	14
2.4	Microservices Patterns	15
2.4.1	Database Per Service	15
2.4.2	Access Token	16
2.4.3	API Gateway	16
2.5	Development Technologies	17
2.5.1	Frontend Technologies	17
2.5.2	Backend Technologies	20
2.5.3	Cloud Service Providers	22
2.5.4	Rich Text Editors	24
2.6	Football APIs	27
3	Requirements	29
3.1	Actors	29
3.2	User Stories	29
3.3	Functional Requirements	30
3.4	Non-functional Requirements	31
4	Architectural Decisions	35
4.1	AWS Cloud Services	35
4.2	C4 Model	36
4.2.1	Context Diagram	36
4.2.2	Container Diagram	37

4.2.3	Component Diagrams	40
4.3	Entity–Relationship Diagrams	46
4.3.1	Football Information Diagram	47
4.3.2	Content Manager Diagram	48
4.3.3	User Information Diagram	49
5	Methodology	51
5.1	Traditional vs Agile Methodologies	51
5.2	Chosen Methodology	52
6	Planning and Risks	55
6.1	Planning	55
6.2	Risk Analysis	59
6.2.1	First Analysis	59
6.2.2	Second Analysis	64
6.2.3	Final Analysis	67
7	Development	69
7.1	Development Process	69
7.1.1	Process Organization	69
7.1.2	Task Organization	70
7.2	Scope Redefinition	71
7.3	Development Timeline	72
7.4	Developed Functional Requirements	74
7.5	Developed Architecture	76
7.5.1	Developed C4 Model	76
7.5.2	Developed Entity-Relationship Diagram	84
7.6	Platform Functionalities	86
7.6.1	Authentication	86
7.6.2	Profile	87
7.6.3	Reviews	89
7.6.4	Review Creation	92
7.6.5	Matches	93
7.6.6	Team	95
7.6.7	Homepage	96
8	Testing	99
8.1	Unit Tests	99
8.2	Integration Tests	100
8.3	Non-functional Tests	101
8.3.1	Security - High Priority	102
8.3.2	Availability - Medium Priority	106
8.3.3	Performance - Medium Priority	106
8.3.4	Scalability - Low Priority	107
8.3.5	Usability - Low Priority	108
9	Conclusion	111
9.1	Conclusion	111
9.2	Future Work	112

Appendix A	User stories	127
Appendix B	Functional Requirements	147
Appendix C	Functionality Showcase	151
Appendix D	Integration Tests	169

Acronyms

AWS Amazon Web Services.

CF *Código Futebolístico.*

CQRS Command Query Responsibility Segregation.

CSP Cloud Service Provider.

DT Development Team.

EC2 Elastic Compute Cloud.

ELB Elastic Load Balancer.

GP GoalPoint.

LE *Lateral Esquerdo.*

MVP Minimum Viable Product.

PO Product Owner.

R Reddit.

RDS Relational Database Service.

RTE Rich Text Editor.

S *Spielverlagerung.*

S3 Simple Storage Service.

SM Scrum Master.

SPA Single-Page Application.

TFA Total Football Analysis.

UI User Interface.

List of Figures

4.1	Context Diagram	37
4.2	Container Diagram	39
4.3	Single-page application component diagram	41
4.4	Football Information microservice component diagram	43
4.5	Content Manager microservice component diagram	44
4.6	User Information microservice component diagram	45
4.7	API Gateway component diagram	46
4.8	Football Information entity-relationship diagram	48
4.9	Content Manager entity-relationship diagram	49
4.10	User Information entity-relationship diagram	49
5.1	Scrum flowchart [112]	53
6.1	First semester	56
6.2	Second semester	58
6.3	Risk Matrix - First Analysis	63
6.4	Risk Matrix - Second Analysis	66
7.1	Linear sprint cycle board	71
7.2	Internship development timeline	73
7.3	Developed context diagram	77
7.4	Developed container diagram	78
7.5	Developed Single-Page Application component diagram	79
7.6	Developed Football Information component diagram	80
7.7	Developed Content Manager component diagram	81
7.8	Developed User Information component diagram	82
7.9	Developed API-Gateway component diagram	83
7.10	Football Information developed entity-relationship diagram	84
7.11	Content Manager developed entity-relationship diagram	85
7.12	User Information developed entity-relationship diagram	86
7.13	Login page	87
7.14	Registration page	87
7.15	Profile - My Data tab	88
7.16	Profile - Reviews tab	88
7.17	Profile - Collections tab	89
7.18	Review list page	90
7.19	Review page - review content	91
7.20	Review page - comments and suggested reviews	92
7.21	Create review page	93

7.22	Create review toolbar expanded	93
7.23	Match list page	94
7.24	Match page	95
7.25	Team page - Matches tab	96
7.26	Team page - Reviews tab	96
7.27	Authenticated homepage	97
8.1	Unit test - Get Team by Id	100
8.2	OWASP ZAP automated scan results	104
C.1	Login page	151
C.2	Registration page	152
C.3	Password recovery page	152
C.4	Password recovery page 2	153
C.5	Public Profile - Reviews tab	154
C.6	Public Profile - Collections tab	154
C.7	Private Profile - My Data tab	155
C.8	Private Profile - Reviews tab	155
C.9	Private Profile - Collections tab	156
C.10	Private Profile - Create Collection Modal	156
C.11	All Reviews page	158
C.12	Review - Title, content, and details	159
C.13	Review - Comments and suggestions	160
C.14	Review - Edit comment	160
C.15	Create review - Choose Match Modal	161
C.16	Create review - Expanded editor toolbar	161
C.17	Create review - Review editor	162
C.18	Create review - Preview review	162
C.19	All Matches page	163
C.20	Match page	164
C.21	Club page - Matches tab	165
C.22	Club page - Reviews tab	165
C.23	Homepage for unauthenticated users	166
C.24	Homepage for authenticated users	167

List of Tables

2.1	Feature list	7
2.2	Content list	7
2.3	Feature comparison	8
2.4	Content comparison	8
2.5	Usage Comparison of Frontend Technologies [35]	17
2.6	Comparison of Frontend technologies	19
2.7	Comparison of Backend technologies	22
2.8	Comparison of Cloud Service Providers	24
2.9	Comparison of Rich Text Editor's popularity	25
2.10	Comparison of Rich Text Editors	26
3.1	Authentication Requirements	31
3.2	Security - Authentication scenario	32
3.3	Security - Authorization scenario	32
3.4	Availability scenario	33
3.5	Performance scenario	33
3.6	Scalability scenario	34
3.7	Usability scenario	34
6.1	Development risks - first analysis	60
6.2	Platform risks - first analysis	61
6.3	Likelihood categories	62
6.4	Impact categories	62
6.5	Development risks - second analysis	65
6.6	Platform risks - second analysis	66
7.1	Completed requirements - Authentication and Profile	74
7.2	Completed requirements - Content and Football	75
8.1	Integration test results - Authentication module	101
8.2	Security NFR - Unauthenticated scenario test results	102
8.3	Security NFR - Unauthorized scenario test results	103
8.4	Performance test results	107
8.5	Scalability test results	108
B.1	Authentication functional requirements	147
B.2	Profile functional requirements	147
B.3	Content functional requirements	148
B.4	Football functional requirements	149

D.1	Authentication module integration test results	169
D.2	Personal Profile module integration test results - Part 1	170
D.3	Personal Profile module integration test results - Part 2	171
D.4	Public Profile module integration test results	172
D.5	Reviews module integration test results - Part 1	173
D.6	Reviews module integration test results - Part 2	174
D.7	Matches module integration test results	175
D.8	Teams module integration test results	176
D.9	Homepage module integration test results	177

Chapter 1

Introduction

This internship consists in the creation and development of a new platform to centralize, create, and interact with football content called Golden Goal. This will be done under the orientation of Prof. Nuno Antunes, PhD. and Prof. Pedro Abreu, PhD. from the Department of Informatics Engineering, and Engr. Arnaldo Moura, BSc, from the company Grama.

This document was created within the scope of the course of Dissertation / Internship in Software Engineering of the Master in Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

This chapter works as a contextualization of the internship. Section 1.1 describes the company and the project while Section 1.2 goes over the project's context and motivation. Following this, Section 1.3 describes the project's objectives and, to finalize the Chapter, Section 1.4 presents the document's structure.

1.1 Grama and Golden Goal

Grama [1] is a company located in Coimbra, founded in 2017, that offers various services for any stage of a software product's life cycle, from its creation and design to the development and management.

Golden Goal is a new project from Grama that is currently in its inception. It was born from a need to centralize the various reviews and football content pieces scattered around the internet into one single platform. At the moment, this content is usually published on an author's personal website or on their social media, which makes it hard to keep track of authors and popular reviews. It will feature three main types of content: football **matches**, match **reviews**, and personal user **experiences** while attending matches at a stadium.

As a football-related platform, it will require structured planning, research, and development to handle the variety and detail of football data. It will be equally important to create a platform that is able to grow and sustain a user base.

Considering this, Grama created an internship proposal to make a Minimum Vi-

able Product (MVP) capable of fulfilling these needs.

1.2 Context and Motivation

As previously mentioned, reviews and opinions are spread throughout the entire internet, making it difficult to find a review about a particular match or team you support or follow. Different authors publish their reviews on their own websites or on popular social media like Facebook [2], Twitter [3], or LinkedIn [4].

This leads to various different problems. Firstly, these reviews are frequently published on websites that are not intended for or specialized in football content, leading to the loss of details and context that can only be provided with a structured database of football information. Secondly, the user's capacity to interact with the content and its authors is very limited in most cases, leading to the best content and authors being hard to discover.

The Golden Goal platform tries to solve the first problem by associating the user-generated content with a particular match. This will allow readers to quickly understand the match's context, the teams involved, the final score, and other meaningful information. The second problem, discoverability, is solved by allowing users to create their own content in the platform and having researched rating systems adapted to each type of content. This would make it simpler to understand which content is popular and allow users to follow other users, letting them receive updates when their favorite authors release new content that they might consider interesting. Users can also create collections of their favorite content and matches to share with each other.

1.3 Objectives

The objective of this internship was to create and develop Golden Goal, a football content-creation platform that aims to centralize football content such as matches, reviews, and experiences. The design for the platform had been previously created by a designer from the company, so the student only developed the backend and frontend components. These components were hosted on the cloud, as described in this internship's proposal.

This internship was divided into two different phases, one for each semester of its duration. The first phase involved researching and analyzing Golden Goal's competitors, the state of the art of some relevant features, and the technologies and methodologies that were used. It also consisted in defining the project's requirements, architecture and analyzing the associated risks.

The second phase consisted of the platform's development, which occurred during the entire second semester of the internship. This was followed by a testing phase, which were made to ensure that the final product was meeting the requirements defined during the first development phase.

To validate the features selected and their correct development, professional football analysts and other specialists that often create and read reviews were consulted. As they have the most experience in the field, their feedback was a good indication of the validity of the features developed and of possible improvements that could be made to the platform.

1.4 Structure

This document is divided into 9 different chapters, created to present and explain the progress and decisions made during the first semester of the internship, as can be seen in the following list:

- **Chapter 1**, presents the context, motivation, and objectives of the internship;
- **Chapter 2** starts by presenting the competitor analysis. After it the state of the art of the technologies researched is introduced, including research about the rating systems that will be used in the project. Then, a comparison of monolithic and microservices architectures is presented, followed by a research about commonly used microservices patterns. To finish the chapter, the backend and frontend technologies are analyzed and chosen, preceding the choice of external Application Programming Interfaces (APIs) used to obtain football information;
- In **Chapter 3**, the project's requirements are elaborated. The chapter begins by presenting the actors that interact with the platform, followed by the creation of user stories and epic stories. To finalize, both functional and non-functional requirements are introduced and prioritized;
- **Chapter 4** describes the architectural decisions taken in this internship. It starts by presenting the services that will be used, which belong to Amazon Web Services (AWS) [5]. This is followed by the C4 [6] model diagrams, describing the architecture, and the entity-relationship diagram, describing the database relationships. These decisions were based on Chapter 2's research and on the project's requirements and needs;
- In **Chapter 5** both agile and traditional methodologies are compared. This is followed by the decision on which methodology is going to be used during the internship;
- **Chapter 6** focuses on the planning and risk management of the project. It starts by presenting the plans made for both semesters, along with their executed timelines. The chapter closes with three risk analyses, the first made at the beginning of the first semester, the second at the end of it, and the third at the end of the internship;
- **Chapter 7** features an overview of the development process of the platform. It describes how tasks were organized and scheduled, alongside which functionalities, requirements, and architectural components were developed;

- **Chapter 8** showcases the tests made to ensure the correct functioning of the platform. It starts by mentioning the unit tests developed, followed by the integration tests and finishing with the non-functional tests;
- Finally, **Chapter 9** concludes the document by providing an overview of the entire internship and its results. This chapter is concluded by providing a possible future direction for the Golden Goal platform.

Chapter 2

State of the Art

The Golden Goal platform needs to handle many different features from user-generated content to football matches. In order to do this, research has been conducted on the state of the art of various topics.

This chapter starts with an analysis of the competition in Section 2.1, focused on comparing all the different features present in the analyzed platforms. This is followed by a research of the rating systems in Section 2.2, a core component of the user interaction on the platform. Related to the architecture, sections 2.3 and 2.4 touch on the topics of Monolithic vs Microservices architectures and Microservices Patterns, respectively. This will be important for the architecture creation in Chapter 4. Finally, this chapter closes with Section 2.5, where the technologies that are going to be used in the project's development are analyzed and chosen.

2.1 Competition Analysis

Analyzing the competition is one of the core parts of creating a platform. It is of the utmost importance to investigate the competitors in the surrounding market and to understand their features and which ones they lack, advantages and disadvantages, and also to find some features that previous planning might have missed.

The elements analyzed are separated into two different categories: website features and website content. The first group is related to generic website features, such as creating accounts, commenting on a post, and being able to search for football matches. The second category is made of the different content that will be available on the website, such as reviews and experiences. All these categories will be explained in a future subsection.

2.1.1 Competitors

Identifying the competitors is the first step in this analysis. As the objective is to create a football-reviewing website with social features, it is important to pick

websites that can handle both of these aspects. Various platforms were inspected and, in the end, six were chosen for this comparison:

- Total Football Analysis (TFA) [7];
- GoalPoint (GP) [8];
- *Lateral Esquerdo* (LE) [9];
- *Código Futebolístico* (CF) [10];
- *Spielverlagerung* (S) [11];
- Reddit (R) [12].

Total Football Analysis is one of the major English-speaking football reviewing websites, with almost 40 000 followers on Twitter [13]. This platform specializes in reviewing tactics and produces exclusive content, such as magazines and podcasts.

The following three platforms: GoalPoint, *Lateral Esquerdo*, and *Código Futebolístico* are popular in Portugal. The first one is known for rating players' and teams' performances after their matches and has a very large presence on social media. The following two are very similar to Total Football Analysis, focusing primarily on reviewing teams and players from a technical perspective. Akin to these two, *Spielverlagerung* focuses on reviews but, in this case, they are written in German.

Finally, Reddit is not a traditional football website but a social media forum. This forum is subdivided into smaller forums or categories called "subreddits", where people post and discuss various topics, from worldwide news to mechanical engineering to, of course, football. This is a very peculiar platform, as it does not produce any type of content. Instead, it relies on its users to populate it with content, either content created within Reddit or external content. This is a very interesting competitor, as it incorporates not only the content aspect from previous platforms but the social aspect as well, which Golden Goal also desires.

2.1.2 List of Analyzed Features

This section introduces the features present on the analyzed websites and the suggested features for Golden Goal. These are separated into two different groups, as mentioned above, with the first group being the website features, represented in Table 2.1, and the second the content features, shown in Table 2.2.

It is important to notice that the content in Golden Goal is user generated, whilst the content in most of the other platforms analyzed - except for Reddit - is produced by the platform's journalists. Therefore, exclusive content will be impossible to produce, bar some exclusive partnerships.

Website Features	
Match database	Users can search for any football match
Mark watched matches	Users can mark matches as watched / not watched
Share to social media	Users can share posts to social media of their choice
Post tags	Users can tag their posts and search by post tags
User accounts	Users can create an account
Verified accounts	Special users have verified accounts
User reviews	Users can create a review of a match, player or team
User experiences	Users can post personal experiences of live matches
User voting	Users can rate reviews and matches
User commenting	Users can comment on reviews and experiences
User lists	Users can create collections of their favorite content

Table 2.1: Feature list

Website content	
Match reviews	Website has match reviews
Historical reviews	Website has reviews of old / historic matches
Team reviews	Website has team reviews
Player reviews	Website has player reviews
Coach reviews	Website has coach reviews
Tactical reviews	Website has tactical reviews
Experiences	Website has articles of people's experiences
Training tips	Website has articles of professional training tips
News	Website has football news
Highlights	Website has match highlights
Exclusive interviews	Website produces and distributes interviews
Podcasts	Website produces and distributes podcasts
Magazines	Website distributes a football magazine
Ratings	Website publishes articles rating players and teams
Games	Website has football games (predictions, fantasy)
Prizes	Website gives away prizes such as jerseys

Table 2.2: Content list

2.1.3 Results and Discussion

The next two tables present the comparison of the competitors analyzed in Section 2.1.1, separated into the two tables introduced in the previous subsection. The green cells marked with the letter "Y" represent a feature the website offers, while the red cells marked with the letter "N" represent the opposite. The "Y*" displayed in the first table marks the features that are present but only for the current season.

Features	GG	TFA	GP	LE	CF	S	R
Searchable match database	Y	Y*	Y*	N	N	N	N
Mark watched matches	Y	N	N	N	N	N	N
Share to social media	Y	Y	Y	Y	Y	Y	Y
Post tags	Y	Y	Y	Y	N	Y	Y
User accounts	Y	N	N	Y	N	N	Y
Verified accounts	Y	N	N	N	N	N	N
User reviews	Y	N	N	N	N	N	Y
User experiences	Y	N	N	N	N	N	Y
User voting	Y	N	N	N	N	N	Y
User commenting	Y	N	N	Y	N	Y	Y
User lists	Y	N	N	N	N	N	N

Table 2.3: Feature comparison

Features	GG	TFA	GP	LE	CF	S	R
Match reviews	Y	Y	Y	Y	Y	Y	Y
Historical reviews	Y	N	Y	Y	N	Y	Y
Team reviews	Y	Y	Y	Y	Y	Y	Y
Player reviews	Y	Y	Y	Y	Y	Y	Y
Coach reviews	Y	Y	Y	Y	Y	Y	Y
Tactical reviews	Y	Y	Y	Y	Y	Y	Y
Experiences	Y	N	N	N	N	N	Y
News	N	N	Y	N	N	N	Y
Highlights	N	N	Y	N	N	N	Y
Exclusive interviews	N	Y	N	N	N	Y	N
Training tips	N	Y	N	N	Y	Y	Y
Magazines	N	Y	N	N	N	Y	N
Podcasts	N	Y	N	N	N	Y	N
Ratings	N	N	Y	N	N	N	N
Games	N	N	Y	N	N	N	N
Prizes	N	N	Y	N	N	N	N

Table 2.4: Content comparison

Starting with the first category, which is website features, it can be seen that sharing on social media and tags are almost universal features in this kind of platform. The exception is *Código Futbolístico*, a website that does not feature post tags. Both of these will be very important to Golden Goal, as sharing to social media can lead to new users joining the platform and tags increase the discoverability of new content.

Implementing a searchable database will be a core feature of the reviews, as it will allow their authors to link them to different matches, and also allow regular users to find reviews of a certain match. All other analyzed websites work only as blogs and therefore do not implement this feature or only have a very limited version of it. Considering this, it is understandable that none of the platforms analyzed allow their users to mark watched matches. The platforms that fully implement a database are usually only focused on that and do not have the content features mentioned. Consequently, they are not being included in this analysis.

In this category, a lack of features related to user interaction can be observed. Few websites allow users to create accounts and the ones that do only allow users to comment on the reviews. Reddit is an exception to this, implementing almost the same features as Golden Goal. It centers its platform around users and user-generated content, something that Golden Goal will aim to replicate and apply to this specific domain.

Comparing the content on all websites, it can be seen that all of them allow for every kind of review, with Total Football Analysis and *Código Futbolístico* both missing historical reviews. However, it is my impression that it was just a matter of these reviews not being made and not the website being unable to support them, as these would be similar to match reviews, which all platforms support.

Total Football Analysis and *Spielverlagerung* emphasize making content for its platform, distributing podcasts, exclusive interviews, training tips, and monthly magazines about the latest news and most popular matches. These are very creative ideas for the future of the platform but are currently hard to implement, considering that the platform is going to focus on user-generated content.

GoalPoint focuses on the latest matches and news, providing match highlights and ratings. It also features prediction-based games that reward winners with prizes and regularly give away football jerseys. All of these are interesting but deviate from the objective of focusing on reviews.

Lateral Esquerdo and *Código Futbolístico*, as it is planned for Golden Goal, only focus on reviews, with the latter also having training articles. However, Golden Goal will differ from these two in the social aspect, being that these work as blogs, and our platform will allow users to interact with the content in more diverse ways.

Reddit also features all the content planned for Golden Goal and even includes some extra topics like news, highlights, and training tips, which may be useful to incorporate into the platform after the Minimum Viable Product (MVP). However, Reddit is a generalist forum and not one made for this specific topic, therefore missing specific features like a football match database.

To conclude, this analysis proved itself to be useful in learning Golden Goal's advantages, allowing users to interact with content such as reviews and experiences on a football-specific platform. It has also shown that, even though most football platforms neglect user interaction, they also produce diverse content that might be interesting to consider for the platform in the future, such as magazines and training tips, even though they require dedicated journalists to create them.

2.2 Rating Systems

Golden Goal will rely heavily on user voting and interaction, as all social media and forums do. It is how users choose what they want to see and the content they value most on the platform.

It is important to find the optimal design for the rating system, as every website has its content rating mechanism - Twitter [3], Facebook [2], and Youtube [14] have the like button, Reddit [12] has the upvote/downvote, and IMDb [15] has the 0-10 stars system. Each of these has its merit and specific uses.

There are three types of ratings systems on the Golden Goal platform: review ratings, match ratings, and experience ratings. All of these present different types of needs and challenges.

The first two will have to face diversity and review bombing problems, as these will be naturally caused by the vastly different number of supporters for each team. While it is important to focus on main teams, it is equally healthy for the platform to show team diversity and article variety to attract a larger audience. It is also crucial that the same two rating systems can handle review bombing, that is, rating a review or a match based on external factors or preferences, not the quality itself.

The experiences' rating system will need to offer authors an option to differentiate their experiences. Considering that the only rating in the experiences is the one given by the author of the experience, diversity and review bombing will not be at play here.

2.2.1 Analyzing a Rating System

The first step to designing a rating system is to understand what a rating system is made of and what can influence the users' votes in a rating system. Investigators from the University of Turin [16] decompose a rating system into four categories:

- **Granularity** - the number of positions on a scale;
- **Numbering** - the numbers associated with each position;
- **Visual metaphor** - star, slider, like button, thumbs up / down, among others;

- **Presence of a neutral position** - even / odd number of positions.

All of the categories mentioned above are combined into what they call the "personality of a rating system". This personality is one of the three elements that the same investigators determine to influence user ratings, which are:

- The item being rated;
- The personality of the user;
- The personality of the rating scale in use.

The first two points are pretty straightforward - different items will have different classifications, and a more optimistic user will usually rate items more favorably than a pessimistic one. The third category is the most interesting out of all of these, as it is the one developers can easily change and influence.

2.2.2 Comparing rating systems

As rating systems have different advantages and disadvantages, it is important to compare all the four components that make their personality.

Starting with the presence of a neutral position, it is reported that not having one on the scale induces bias in the results. So, in order to avoid this, a middle point is recommended. Having negative numbers on the scale also leads to an alteration in the rating, inflating its results.

Taking into account the visual metaphor, both thumbs ratings (unary and up-vote/downvote) and the star ratings seem appropriate for the platform. The former tends to evoke feelings related to human behavior while the latter is easily associated with hotel ratings or popular restaurant rating applications like Tripadvisor [17].

Rating systems with higher granularity lead to fewer votes but are able to gather more information. The exception is the unary system, which leads to fewer ratings than all other systems, due to only having one option. Lower granularity rating systems tend to be faster and simpler [18].

In terms of user preference, the slider is the least preferred, followed by the unary, the thumbs up / down system, and the most preferred, the star rating [18].

A study compared both unary and thumbs up / down systems and evaluated the impact of the removal of the downvote button [19]. It used both platform-level and user-level analyses, where the former dive into the consequences for the platform as a whole and the latter inspect how it affected the various user groups registered in the platform before the change.

The **platform-level analyses** showed that:

1. The number of posts increased;

2. The number of replies increased;
3. The content of the posts became richer and more diverse;
4. The toxicity of replies decreased.

The **user-level analyses** showed that:

5. The number of replies per user increased;
6. The number of posts per active user (who had posted previously) did not change;
7. Newer users were more affected by the change, while older users were less affected;
8. Light users became less toxic after the change;
9. Heavy users became more toxic after the change.

Considering the overall effect on the platform, this study has shown very powerful results. Results number 1 and 2 are of utmost importance for all social websites, as user content creation and interaction are the core of these platforms. Higher amounts of content will attract more users, becoming a vicious circle.

These results can be linked directly to results 5 and 6, respectively, where it can be concluded that the removal of the downvote button increased the discussion on the platform and that users that did not post before are now doing it, as the average number of posts per active user did not change but the number of posts increased.

The results of the third topic are very interesting and meaningful, as content diversity is one of the problems identified in the soccer content of Reddit, Twitter, and other social media. Everything that can be done to increase the variety of content is extremely positive, especially considering that this area is particularly biased toward the biggest teams. The smaller teams are, most of the time, equally as interesting, but forgotten in the midst of giant international names.

Finally, while the study has concluded that the toxicity of replies decreased in topic number 4, it is important to analyze points number 8 and 9 too. This shows that, as more light users were participating in the discussion, they tended to reduce the average toxicity in the replies, though the heavy users compensated for the lack of the downvote button with aggressive replies, as expected. This is a negative point - especially considering that Golden Goal is a football website - but it is also fixable with added moderation to avoid problems that are naturally filtered by the downvote button, such as spamming, verbal abuses, racism, and xenophobia, among other unwanted behaviors. An interesting idea is not only to remove the downvote button but to replace it with a report button, as it will be easily and naturally accessible, just like the downvote, and it will help with filtering unwanted behavior.

Topic number 7 is not particularly interesting, as it does not apply to an unreleased platform.

2.2.3 Review Bombing

The second main problem Golden Goal's rating system may face is review bombing. Review bombing is a phenomenon where groups of people, sometimes using multiple accounts, attack the rating of a certain product or item, giving it the lowest possible review. This is prevalent in review aggregator websites such as IMDb, Rotten Tomatoes [20], and Metacritic [21], where movies, series, albums, and video games are rated and ranked.

These negative reviews are not related to the quality of said product but are made with harmful intent, trying to lower its sales and popularity. They differ from natural extreme reviews because they are usually motivated by an ideological disagreement with the product or the people related to it. In Golden Goal's case, the most likely source of review bombing will be linked to a team's supporters rating negatively a match that their team lost or an article that criticizes their team.

It is therefore important to handle these cases and ignore or undervalue ratings that come from review bombers. There are a few characteristics that can be exploited to identify these users. Investigators from the University of Catania [22] found that anomalies in the textual content and the usernames can be key to detecting fake reviews. These negative reviews are usually paired with rude language in their textual content and are negatively correlated with reviewing experience. Review bombing also tends to occur in peaks, usually close to the release of the product or near a controversy related to it.

So what can be done to ensure the validity of the reviews? Valuing reviews from more experienced users is a good starting point, as they tend to be more reliable and less prone to review bombing. This is something IMDb does in their rating system, as only regular voter's ratings are considered in the "Top X movies or series" lists [23] and IMDb's help website states that "IMDb publishes weighted vote averages rather than raw data averages" [24], even though they do not disclose the filters applied to the data. It is also recommended that an the visibility of an article or match is not only determined by its rating but also has into consideration comments, recency, and other relevant factors.

Detecting and flagging accounts that are constantly voting against or for a certain team as well as accounts whose votes coincide with a team's results is important, as they are most likely not evaluating the articles or the matches themselves.

There are some simpler ways to try to avoid review bombing. The first is to simply remove the thumbs-down button from the binary thumbs system. A less effective way is to hide the overall rating during the first hours of posting, when review bombing is more likely to occur. This can be taken to an extreme by disabling rating for a few hours after a match has ended or an article has been posted, but this is not as interesting in Golden Goal's case, as it will lead to a huge loss in user interactions.

Finally, it is important to note that some of these ideas will be more useful in the future, as the platform grows. Review bombing will most likely not be a problem with a small user base.

2.2.4 Analysis

As a conclusion to this section, the gathered information is going to be applied to the platform's three rating systems.

The reviews' and matches' rating systems are similar in terms of their needs, mainly facing diversity and review bombing problems. Considering this, the focus was on the thumbs up / down systems, as these are very commonly used in forums, due to being faster for users to interact with. It was also decided to use the unary version of this system, removing the thumbs down button from it. This benefits content diversity, generates more posts, promotes discussion, and controls the review bombing problem, as users will not be able to negatively vote on a rating. The increased toxicity the unary system brings might become a problem after the platform grows, but this is easily solvable by increasing the moderation or even by implementing a reporting system.

Finally, for the experiences' rating system, the choice was the star rating system. As this system means to differentiate between each experience, granularity is the most important characteristic the rating system must have. In this case, review bombing and diversity will not be a factor, as these are only rated by the author and not the general public.

2.3 Monolithic vs Microservices Architectures

One of the most important choices in the development of a software project is deciding the architectural style. The debate of Microservices vs Monolith is very interesting, as both have their pros and cons, and making the right decision often depends on various factors.

A Monolithic architecture is an architecture where the system is built and deployed as one, large, single unit, which handles everything from the user interface to the business logic and the database. This offers several advantages, such as it being simpler to build, test, and deploy, as it has fewer cross-cutting concerns. It also offers better performance, by not having to contact multiple services to fulfill a request, and has less operational overheads [25, 26].

A Monolithic architecture can have its downsides, as everything is crammed into one unit. There exists a problem with the reliability of the service when any of its modules fail, as it can bring the whole application down. The Monolithic structure also forces the use of a single technological stack for the entire unit, as everything is tightly coupled, which makes it hard to introduce new technologies. It also creates the need to redeploy the entire application to apply updates and makes it impossible to scale the parts of the application that are under heavy load [25, 26].

As an answer to the tightly coupled Monolithic architecture comes the Microservices architecture, based on small and independent services that communicate with each other. This offers incredible flexibility, as it is possible to: add new

technologies to a service's stack without interfering with other services, scale each service independently as needed, and deploy updates without having to redeploy the entire application. It also helps with the reliability issue, as the problems of each service are isolated. Furthermore, in cases with bigger development teams, it is possible to assign a specific team to develop certain microservices. These teams will be easier to integrate, as they will only need to understand the microservices they are developing [25, 26].

The downsides of the Microservices architecture come from the isolation of each service. It is now necessary to handle previous inexistent cross-cutting concerns, such as logging and health checks. The services need to communicate, therefore leading to slower performance, higher operational overhead, and security threats. Testing and debugging also become harder the more moving parts are added to the system. Microservices also bring additional complexity to processing and finding information, as the microservices need to communicate to find the information separated throughout them [25, 26].

For this project, while it is clear that both architectural styles have their advantages and disadvantages, the final choice is a Microservices architecture. It will allow the development and easy deployment of independent modules and the addition of extra modules to it in the future while keeping the platform available and reliable.

2.4 Microservices Patterns

To correctly develop a Microservices architecture it is important to understand the different patterns that can be implemented and used. These patterns usually answer questions such as "Where is data stored?" and "How do services communicate with each other?", and can be separated into different categories such as cross-cutting concerns, testing, observability, data, and deployment, among others [27].

In this section, three different patterns are going to be analyzed: Database Per Service, Access Token, and API Gateway, as these are the most relevant for the development of the project and some of the most used by the company.

2.4.1 Database Per Service

One of the first questions that come to mind when considering a Microservices architecture is "where is the data going to be stored". To answer this question, there are two different patterns, as you can either have a database per service [28] or a shared database for all services [29].

While having a shared database can help guarantee data consistency, this goes against the idea of having loosely coupled microservices, undermining many of the benefits of this architecture.

If the architecture follows the database per service pattern, all of the services will be independent of each other. This can be achieved using the following methods, ordered from closely coupled to loosely coupled:

- Private-tables-per-service: Each service with its own tables within the shared database;
- Schema-per-service: Each service with its own schema within the shared database;
- Database-server-per-service: Each service has its own separate database server.

The separation of data between services creates the need for synchronization between databases, which can be achieved using the Saga pattern [30]. This updates all services using a transaction log, which is updated by all microservices. This way, changes are propagated across services while maintaining consistency.

There can also be the need to gather data from multiple services. This can be solved using patterns like API Composition [31] and Command Query Responsibility Segregation (CQRS) [32].

API Composition creates a service, like an API Gateway, which performs a join operation by querying databases to aggregate data from multiple services.

The CQRS pattern enables the creation of denormalized views of the databases, which are pre-calculated representations of the data. These are stored separately from the main databases and can be used to gather data from multiple services efficiently, eliminating the need for complex joins and processing during queries.

2.4.2 Access Token

The access token is a widely adopted pattern that plays a crucial role in verifying the identity of a client making a request, which is especially important when handling sensitive or valuable information.

This control is ensured by the usage of an access token, emitted by the backend, which identifies the client as authenticated. This token is assigned to the client and can be included with subsequent requests to validate their authenticity [33].

The utilization of access tokens is an effective mean of controlling access to protected / private resources, as it ensures that only authorized clients with valid tokens are able to access them.

2.4.3 API Gateway

When a user accesses a microservices-based service, he needs his requests to be sent to the correct microservices to be processed. However, it is impractical for the client to need to determine where the adequate microservices are located.

The solution to this problem is the use of an API Gateway, which acts as a single point of entry to the backend and redirects the requests to the correct microservice. This API Gateway can also guarantee that the client is authenticated, through the use of an Access Token, and can adapt the requests to the client's needs, as a traditional browser client may need different information than a mobile client. Additionally, having a single point of entry to the backend helps with security problems, as it can protect it from every request made [34].

2.5 Development Technologies

This section presents the technologies analyzed for the development of the platform, alongside their advantages and disadvantages. It is divided into three subsections: Frontend technologies, Backend technologies, and Cloud Service Providers. In each of the mentioned subsections, there will be a presentation of each technology considered. This will be followed by a comparison of all the technologies presented and finalized by choosing the technology which better fits the development of the platform.

2.5.1 Frontend Technologies

The first step in the analysis of Frontend technologies is to find the most popular technologies in the market.

The State of JS [35] is an annual survey in which the authors analyze the progress and usage of various libraries, tools, and frameworks related to JavaScript. In the most recent edition, the study had 16085 respondents and showed the results represented in Table 2.5.

	2019	2020	2021
React	80%	80%	80%
Angular	58%	56%	54%
Vue.js	47%	49%	51%
Svelte	8%	15%	20%
Preact	12%	14%	14%
Ember	12%	11%	9%
Lit	-	5%	7%
Alpine.js	-	3%	6%
Solid	-	-	3%

Table 2.5: Usage Comparison of Frontend Technologies [35]

As can be seen from Table 2.5, which represents the percentage of users that have experience with a certain language, there were three technologies separated from the rest: React, Angular, and Vue.js. Due to their natural relevance in the market, these were chosen to be analyzed for this internship. Additionally, Svelte showed

the largest rise in popularity out of all the technologies, resulting in the following list of Frontend technologies:

- React [36];
- Angular [37];
- Vue [38];
- Svelte [39].

React

As shown in Table 2.5, React is currently the most used Frontend JavaScript library. Released in 2013 by Meta [40], this library features reusable components and can be used to build user interfaces.

It is a simple and easy-to-learn library that features a high diversity of tools and a large community and ecosystem. Its main feature is the use of a virtual DOM, which enhances the performance of the website's rendering, controlling updates and refreshing only when needed [41, 42, 43].

React's main disadvantage is a result of its constant evolution, as its documentation struggles to keep up with the rate of updates. While being easy to learn and use, its initial learning curve can be steep due to the use of JSX to write HTML in React [41, 42, 43].

Vue.js

Vue.js is a JavaScript framework that follows React with the second highest popularity. Released a year after, Vue.js also features a virtual DOM and is component-based [4][5]. It provides its users with extensive and detailed documentation accompanied by a simple and clear syntax. This makes Vue.js one of the simplest frameworks to learn [41, 42, 43].

Whilst really popular in China, it has still has to catch up with other frameworks' popularity in other regions. This creates a language barrier that makes it difficult to interact with the community and to read parts of the documentation [41, 42, 43].

Angular

Angular is a component-based web development framework that was released in 2016 by Google. Its documentation is heavily detailed and the framework provides high-speed performance through the usage of an in-built two-way data binding, synchronizing the view and the model [41, 42, 43].

Angular presents two main disadvantages, as it is a hard framework to learn and it presents a smaller ecosystem and community than Vue and React [41, 42, 43].

Svelte

Svelte, released in the same year as Angular, is one of the fastest-growing Frontend frameworks. Its appeal lies in its speed and its lightweightsness, as it differs from the other frameworks by compiling the code instead of placing that burden on the browser [41, 42, 43].

However, even with its growing community, there is still a small ecosystem and a lack of tooling and tutorials that do not affect more popular frameworks [41, 42, 43].

Comparison

The advantages and disadvantages of each technology mentioned can be seen in Table 2.6, alongisde extra information about them, such as compatible languages and the number of GitHub stars [44] (data taken on 19-10-2022).

	Frontend			
Framework	React	Vue	Angular	Svelte
Release Date	2013	2014	2016	2016
Languages	JavaScript TypeScript	JavaScript TypeScript	TypeScript	JavaScript TypeScript
Star Count	192 000	198 000	83 000	61 100
Advantages	Component based Virtual DOM Stable Diversity of tools Easy to learn Vast community	Component based Virtual DOM Fast Clear syntax Easy to learn Detailed documentation	Component based Two-way data binding Detailed documentation	Improved reactivity Lightweight Simple Fast
Disadvantages	Less documentation Steep learning curve	Lack of features Language barrier	Hard to learn Smaller ecosystem	Smaller ecosystem Lack of support Lack of tooling

Table 2.6: Comparison of Frontend technologies

Decision

Out of all the technologies presented, the final choice ends up being a battle between React and Vue.js, as Angular presents a steep learning curve and Svelte is still facing the growing pains of having a smaller community and ecosystem.

Between the two final contestants, React ends up with the upper hand, as it has the largest ecosystem of the two. It also has its documentation and tutorials fully written in English, compared to the language barrier with the Vue.js community.

The company also offers support in both languages but the student only has experience using React. This is a major advantage for React, as the student will not have to learn a new language from scratch.

2.5.2 Backend Technologies

As done in the previous section, the first step in choosing a Backend technology is finding which are the most used. Backend frameworks that use languages in which the student has no experience, such as PHP and Ruby, will be excluded.

Based on their popularity [45, 46], deduced each technology's star count, the frameworks that will be analyzed are:

- Spring Boot [47];
- Express.js [48];
- Django [49];
- Flask [50].

The Quarkus [51] framework, due to its rapidly increasing popularity [52], will also be added to the list of frameworks to analyze.

Spring Boot

Released in 2003, this is the oldest Backend framework still at the top of the popularity charts. Using Java, this framework is stable and incredibly versatile, due to its functionalities and integrations. It features multi-threaded performance and high security, accompanied by detailed documentation [53]. It also provides its users with a quick setup process [54].

The disadvantage of Spring Boot is one of the downsides of its versatility. Unused dependencies can increase the size of the deployable and decrease its performance [53, 54]. It is also slower than Quarkus, another Java-based framework [55].

Quarkus

Quarkus is a Java-based framework that was released in 2019 by RedHat. This framework's advantage lies in its speed, as it is way faster than Spring Boot, having a faster boot time and performing faster reloads [53].

However, as a newer framework, its ecosystem is still small and its documentation, while detailed, is still limited. Additionally, it also presents a harder installation than its java counterpart [53].

Express.js

Express.js is a framework that runs on Node.js and, therefore, uses JavaScript or TypeScript. Because of this, it is easy to learn and it has a developed ecosystem. It also provides good scalability and efficient memory usage [55] [56] [57].

Its downside comes from it being a single-threaded framework. Because of this, it performs poorly, especially with CPU-blocking calls [55, 56].

Django

Django is a very popular open-source Python framework. It comes with many features implemented out of the box, accompanied by very detailed and thorough documentation, which leads to a fast development. It also is a very scalable, versatile, and secure framework [58, 59, 60].

However, it has a very monolithic structure associated with pre-defined variables and files, making it consume unnecessary resources and perform slower than other frameworks [59, 60].

Flask

Flask, unlike its Python counterpart, is a very lightweight and flexible framework. While simple, it is also expandable, making it a scalable framework. It also has lots of detailed documentation on its official website [61].

Unfortunately, it has a way smaller community than Django, leading to an inferior ecosystem with a lack of tools and tutorials.

Comparison

Similar to the previous section, the following Table 2.7 is going to present the comparison between all five technologies, adding the compatible languages and the GitHub [44] stars (data taken on 19-10-2022).

	Backend				
Framework	Spring Boot	Quarkus	Express.js	Django	Flask
Release Date	2003	2019	2010	2005	2010
Languages	Java Kotlin	Java Kotlin	JavaScript TypeScript	Python	Python
Star Count	49 600	43 000	58 700	67 000	61 000
Advantages	Quick setup	Fast boot time	Low memory consumption	High security	Flexibility
	Multi-threaded		Good scalability	Scalability	Scalability
	High security	Fast hot reloads	Easy to learn	Versatility	Lightweight
	Detailed documentation	Detailed documentation	Developed ecosystem	Detailed documentation	Detailed documentation
Disadvantages	Increased project size	Complicated installation	Low performance with blocking CPU calls	Resource hungry	Small community
		Small community		Monolithic	Small ecosystem

Table 2.7: Comparison of Backend technologies

Decision

After analyzing all five previous technologies, Spring Boot ends up at the top. Quarkus and Flask both had way smaller communities which would create unnecessary difficulties during the project, while Express.js and Django presented some problems related to their performance and overhead, respectively.

Spring Boot is the most versatile and stable out of all frameworks analyzed, and features detailed documentation and a fast setup to support the quick development of this project. It is also used in various projects made by the company, which will help the student get quality feedback during the internship.

2.5.3 Cloud Service Providers

Choosing a Cloud Service Provider (CSP) is equally as important as choosing the technologies that will be used to develop the platform.

Currently, the main CSPs are Amazon, with Amazon Web Services (AWS) [5], Microsoft with Azure [62], and Google with Google Cloud [63], having respectively 34%, 21%, and 10% of the cloud market share. The remaining CSPs all have under 5% of the market share [64].

In this section, the three main CSPs mentioned above are going to be discussed in order to understand the advantages and disadvantages of each one.

Amazon Web Services

AWS dominates the CSP market by offering an enormous variety of services and features.

Founded in 2006 by Amazon, it is one of the most developed CSPs, having a worldwide network of data centers serving 245 countries and 25 geographic regions. These centers are recognized for their reliability, speed, and availability. AWS is a generalist service, offering services from every technological area but not specializing in any of them [65, 66].

The main downside is the cost, as expenditures can be considerably high, even with their pay-per-use model, which are sometimes complex and hard to understand [65, 66].

Azure

The most recent of the three CSPs analyzed, Microsoft's Azure has the second largest market share.

With quarterly updates, Azure also offers a great variety of technologies, even if lower than the offer from AWS. Having partnerships with Oracle, VMware, and SAP, Azure's strength comes from the creation of an ecosystem and connecting various services into one, especially when integrated into Microsoft's already giant ecosystem of products [65, 66].

Compared to AWS, it has lower availability zones and is, unfortunately, equally expensive without providing any guaranteed capacity [65, 66].

Google Cloud

Google Cloud sits on the throne of the most affordable out of the three CSPs analyzed, providing its users with a pay-as-you-go model, a pricing calculator, and free credits for new clients and startups to use. It also offers a worldwide network of data centers comparable to Amazon's, being equally as reliable and available [65, 66].

However, even though Google has been acquiring new services and adding them to the cloud, such as Firebase [67], it is still held back by its specialization in machine learning and data science, which will not be of any use for Golden Goal [65, 66].

Comparison

In a similar fashion to the Frontend and Backend sections, the previous three CSPs will be compared in the following Table 2.8:

	Cloud Service Providers		
Cloud	AWS	Azure	Google Cloud
Release Date	2006	2010	2008
Owner	Amazon	Microsoft	Google
Market Share (%)	34	21	10
Advantages	Worldwide network of data centers Reliability Variety of technologies	Great ecosystem Variety of technologies	Worldwide network of data centers Reliability Affordable
Disadvantages	High prices	High prices Lower availability zones	Limited features and services

Table 2.8: Comparison of Cloud Service Providers

Decision

After comparing the three CSPs, AWS is the clear winner. Google Cloud presents a limited amount of services and its specialization is not an advantage for the platform. Azure, while having a decent amount of technologies and features, has a lower availability for the same price as AWS. In addition to this, Golden Goal will not benefit from Azure's and Microsoft's ecosystem.

AWS has the largest variety of services and is well known for its reliability and availability. It is also already used by the company and the student has already used it before, both things that will help the development of the project.

2.5.4 Rich Text Editors

For the Golden Goal users to successfully create and edit reviews, they will need to use a built-in text editor. This text editor must be able to include images and must be simple for the Golden Goal user to successfully operate. Therefore, it is of utmost importance that the website includes a Rich Text Editor (RTE), a technology that is deserving of being researched.

After gathering the technologies available on the market [68, 69, 70, 71], these were compared based on their number of GitHub stars [44], to find out their popularity (data taken on 20-10-2022). All of the RTEs with more than 5000 stars are represented in the following Table 2.9:

	Number of stars
Quill	34000
Slate JS	25800
Draft.js	22000
Tiptap	17000
Lexical	12000
React Draft Wysiwyg	5900
ReactQuill	5500

Table 2.9: Comparison of Rich Text Editor's popularity

Out of these, Draft.js [72] and Lexical [73] are both from Meta [40]. However, the former is no longer supported and has been replaced by the latter [74]. React Draft Wysiwyg [75] is extremely less popular than all the other alternatives and will not be considered. Quill [76] has a React package - ReactQuill [77] - and both of them will be compared together.

That said, the next four RTEs will be analyzed:

1. Lexical [73];
2. Slate [78];
3. Tiptap [79];
4. Quill + ReactQuill [76, 77].

Lexical

As mentioned, Lexical is the new RTE from Meta. It is a fast editor that presents a great variety of features, allowing users to add diverse content to their text. As it has completely replaced Draft.js in Meta products, it has a rapidly growing community and great documentation. Lexical is adding features at a fast pace and, considering that both React, the chosen Frontend technology, and Lexical are Meta products, they have a really easy setup and work well together [80].

The downsides of Lexical come from being a recent project, as it still has a smaller ecosystem than the other, more mature, options presented [80].

Slate JS

Slate was inspired by Draft.js and, even though it is still in beta, it is capable of processing and featuring complex content. It is very extensible, through the use of plugins, providing the user with good customizability. It also features good and detailed documentation [69, 70, 71].

Its main downside, apart from being in beta and missing mobile support, is that it is harder to learn and set up than the other three options considered [69, 70, 71].

Tiptap

Tiptap is a fully customizable text editor that has a variety of features, including real-time collaboration. It has many keyboard shortcuts to accelerate content editing and features a large community with good documentation [70, 71].

However, like Slate JS, Tiptap is still in beta and has only added official React support very recently [70, 71].

Quill

Quill is a very popular RTE that allows its users to create and format custom content. It features style presets that allow for fast customization of its User Interface (UI). If Quill's React package is being considered, it is also very simple to set up [69, 70, 71].

Quill's problems are the limited customizability it offers and the XSS security vulnerabilities it has. It features few updates and its React package is outdated [69, 70, 71].

Comparison

The following table, 2.10, shows the advantages and disadvantages of each RTE. The last row also includes, in parenthesis, information about Quill's React package, as it has its own merits and shortcomings.

	Advantages	Disadvantages
Lexical	Fast Feature variety Easy setup Maintained and used by Meta Rapidly growing community Good documentation	Recent project Smaller ecosystem
Slate	Complex content and features Supports plugins Good customizability Good documentation	Harder to learn No mobile support Beta version
Tiptap	UI customizability Feature variety Keyboard shortcuts Collaborative editing Large community Good documentation	React support added recently Beta version
Quill (React)	Custom content Formatting support Editor style presets (Easy to set up)	XSS security vulnerabilities Limited customizability (Outdated package)

Table 2.10: Comparison of Rich Text Editors

Decision

After comparing all four RTEs, it is time to decide the most adequate one for the Golden Goal platform. Quill is automatically excluded because of the XSS security problems. Slate is harder to learn and is currently in beta, which may hinder the project's progress.

Finally, between Tiptap and Lexical, Lexical comes out on top, as Tiptap's React support is very recent and it is still in beta, while Lexical is made by the same company as React and features a rapidly growing community to combat its downsides.

2.6 Football APIs

The Golden Goal platform will need football information in order to function, which will be retrieved from external APIs. These APIs need to be free to use and should at least have recent information about the top competitions, matches, players, and coaches.

Only two APIs were able to fulfill these requirements, after an extensive search in RapidAPI [81], a popular API market, and two API lists [82, 83] that compiled sources of football data. These were API-Football and Football-data, both of which worked under a freemium model.

Football-data's free plan [84] allows for 10 calls per minute, having a maximum of 14400 API calls per day, which is significantly higher than API-Football. It provides information on:

- Competitions;
- Teams;
- Match results.

The selection of content obtainable in the free plan is limited, only allowing access to 12 competitions. However, the ones offered are among the most popular, meaning that the data obtainable from this API holds significant value. As these topics make the core of the information needed for Golden Goal, this API will be of great use to the platform.

API-Football [85] has a free plan with no restrictions on the content you can access, being limited only by the number of daily requests, which must not exceed 100. It provides the same information as the previous API, while also offering information on:

- Coaches;
- Players;

- Player statistics;
- Match statistics;
- Match lineups.

The final decision taken was to implement both APIs into the project. Football-data API will be used to get general information about the top competitions, teams, and matches, making use of the higher number of requests allowed, while API-Football will complement it by providing detailed information on coaches, players, matches and statistics.

Chapter 3

Requirements

This chapter defines the functionalities of the Golden Goal platform. It will start by introducing the actors that will interact with the platform in Section 3.1. Then, Section 3.2 describes how Golden Goal works, from the user point of view, using the user stories created for the project. Finally, the following sections, 3.3 and 3.4, terminate the chapter, defining the features of the platform and how it behaves using functional and non-functional requirements, respectively.

As the platform's mockups were created before the beginning of the internship, some of the platform's requirements were extracted from them. However, the student actively collaborated with the Golden Goal team to improve and broaden these requirements. The following sections will present the comprehensive set of requirements that have emerged from this collaboration.

3.1 Actors

The platform will work in a similar way to social media, featuring two types of users: visitors and logged in.

Any type of user can view the content present on the platform, such as teams, matches, reviews, experiences, and author profiles. However, only logged-in users will be allowed to interact with these. Logged-in users will be able to rate matches and reviews, create collections of content, create reviews and experiences, follow authors, and follow teams, among others.

3.2 User Stories

User Stories (US) are a very useful resource used to describe a project's features from the point of view of the final user [86]. They are often used due to their flexibility, as they adhere to the following structure:

As a [user type], I want to [do some action], so that [reason].

This is a very simple structure, focusing on actions and purposes. Because of their focus on the user, they allow for frequent changes and their simple nature allows non-technical users to understand them, fitting nicely into agile methodologies.

Because they represent simple features due to their structure, US are grouped into Epics that describe the theme of the US contained in them [87].

To ensure the validity of these User Stories, a meeting was conducted with a under-21 football analyst working for the Portuguese Football Federation. He is an integral member of the platform's target audience, as he actively engages with and contributes to football review content, a significant aspect of his professional role. During the meeting, he was able to explore and interact with the platform's mockups, which provided valuable feedback and helped validate the platform's design and functionalities.

As there are too many stories to fit in the current section, they are present in their entirety in Appendix A. However, to show how they are structured, the next example presents one US belonging to Authentication Epic:

Epic-1: Authentication

US-1: Register

- (a) **As an unauthenticated user, I want to register myself on the platform so that I can log in to access its functionalities.**

I want to insert:

- Name;
- Username;
- Email;
- Password;
- Password confirmation.

3.3 Functional Requirements

Functional requirements are descriptions of features that a platform can have [88]. As they are written from the point of view of the system, they usually feature a more technical language than the one used in the US, which are written from the point of view of the users [89].

The functional requirements are always prioritized, as it is crucial to understand which ones are vital for the platform and which ones are not as meaningful. To achieve this, Golden Goal's functional requirements will be prioritized using the MoSCoW method. Each capitalized letter of the word MoSCoW stands for a level of prioritization [90], described as followed:

- **Must** - The highest priority - Describes requirements that are non-negotiable and/or mandatory for the platform to work in an intended fashion;

- **Should** - The second highest priority - Describes requirements that, while very important and adding significant value, are not vital for the platform;
- **Could** - The second lowest priority - Describes requirements that are slightly important but are not particularly noticeable if left out;
- **Won't** - The lowest priority - Describes requirements that are not important or valuable for the time frame considered.

In a similar fashion to user stories, the complete list of requirements, separated by areas, is present in Appendix B. However, to show how they are structured, the next example presents the Authentication area requirements:

Area	Id	Requirements	Logged Off User	Logged In User	Priority
Authentication	FR-1	Login	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Must
	FR-2	Logout	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-3	Register	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Must
	FR-4	Recover password	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Must
	FR-5	Change password	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-6	Remember login information	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Could

Table 3.1: Authentication Requirements

3.4 Non-functional Requirements

Non-functional requirements (NFR) are descriptions of how a system should operate and function, with some of the most common NFRs being security, reliability, performance, maintainability, scalability, and usability [91]. These work as restrictions and specifications of a project, as it must adjust and adapt to handle its NFRs [92, 93].

The Non-functional requirements in the Golden Goal project, which are going to be presented next, are defined by a description followed by a scenario. A scenario is made of the following six parts [94]:

- Source;
- Stimulus;
- Environment;
- Artifact;
- Response;
- Response Measure.

Each NFR mentioned will also have its priority defined, as not all NFRs are equally important to the platform. This priority can take three different levels: high, medium, or low.

Security - High Priority

This NFR refers to how capable a system is to protect personal information from users acting under malicious intent [95]. It is of the highest priority, given that any access or change to the user's personal information is a violation of this user's privacy and the European Union's GDPR [96].

In this case, the Security NFR will be separated into two different scenarios, one addressing non-authenticated users accessing information meant for authenticated users (Table 3.2) and one for users accessing unauthorized information (Table 3.3), for example, other user's private information.

	Description
Source	Non-authenticated user
Stimulus	Accesses a page without permission
Environment	Operation in a protected environment
Artifact	System
Response	The access is blocked User is redirected to an error page
Response Measure	The user does not enter the page Data is not compromised

Table 3.2: Security - Authentication scenario

	Description
Source	Non-authorized user
Stimulus	Accesses unauthorized content
Environment	Operation in a protected environment
Artifact	System
Response	The access is blocked User is redirected to an error page
Response Measure	The user does not access unauthorized information Data is not compromised

Table 3.3: Security - Authorization scenario

Availability - Medium Priority

Availability refers to the capacity in which a system can be accessed by its users, usually measured in the percentage of the supposed working time it can ensure [96]. While not as important as security, especially for the creation of an MVP, it is important to consider how it will perform and how it can tolerate failures.

As AWS's services are going to be used throughout the platform, its availability goal is based on it too. Most of the services have an availability of 99.99%, with instances of EC2 and RDS having 99.95% of availability [97].

While the availability of Golden Goal is directly linked to AWS's, it is important to point out that the availability is also affected by the quality and robustness of the platform's code. Therefore, the code will need to be thoroughly tested to ensure it will not negatively affect it.

	Description
Source	Server
Stimulus	The server fails
Environment	Normal operation
Artifact	Server
Response	The system uses other servers to continue operating
Response Measure	The system must ensure 99.95% of availability throughout the year

Table 3.4: Availability scenario

Performance - Medium Priority

This attribute describes how the system performs responding to an event [98]. Similarly to the availability NFR, this is not as important as security, but having a slow performance in a user-focused website can quickly lead to user abandonment due to frustration.

In this case, the focus is on having low latency, as all pages should load in between 1 and 2 seconds [99].

	Description
Source	User
Stimulus	A user requests a page
Environment	Normal operation
Artifact	System
Response	The system returns the page as soon as possible
Response Measure	Page requests must be presented in under two seconds

Table 3.5: Performance scenario

Scalability - Low Priority

Scalability represents the capacity of a system to handle spikes in load while keeping its performance. Systems can scale both vertically and horizontally, adding more resources and more units, respectively [99]. The system should be able to handle load increases of at least 50% of its average load [100].

This NFR has a low priority because, while it is really important for social media to handle load increases, this internship focuses on the creation of an MVP

and not a fully-fledged product, which means it will not have enough users to increase this NFR's priority.

	Description
Source	Users
Stimulus	Multiple users requests in a short interval
Environment	Overloaded conditions
Artifact	System
Response	The load is divided among all units More resources are added to the system
Response Measure	The system must be able to handle load increases of 50% of the average load

Table 3.6: Scalability scenario

Usability - Low Priority

Usability measures how easily can a new user start using a platform and its features. Having good usability and user experience helps attract new users to the platform, as they can now begin using it without experiencing any major problems [101].

This NFR has a low priority as it is important to consider in user-focused websites, but many accessibility features are out of the scope of this internship, such as the "listen to a review" requirement seen in Chapter 3.

	Description
Source	User
Stimulus	Enters the platform for the first time
Environment	Normal operation
Artifact	System
Response	The user is using the platform productively
Response Measure	Average users need under 2 minutes to understand the platform

Table 3.7: Usability scenario

Chapter 4

Architectural Decisions

The present chapter addresses the architectural decisions of the project that is going to become Golden Goal.

This chapter will start by introducing the AWS services [5] that are going to be used in the architecture in Section 4.1. The following section, 4.2, will feature the C4 model, which describes the platform's architecture. It presents the context diagram in Subsection 4.2.1, the container diagrams in 4.2.2, and finishes with the component diagrams in 4.2.3. The chapter ends with the entity-relationship diagrams in Section 4.3, showing the content of the databases and how it is related.

4.1 AWS Cloud Services

As decided in Section 2.5.3, the cloud service provider chosen to host the Golden Goal platform is Amazon's AWS.

Bearing that in mind, it is important to present each of the AWS services that will be used before they are shown in the C4 diagrams. The platform will use the following services:

- **Cognito [102]** - Service used for session, authorization, and authentication management. Provides its own database to keep sensitive information, such as passwords;
- **CloudFront [103]** - Service responsible for distributing static content to the users, like the web app static files;
- **Elastic Load Balancer (ELB) [104]** - Service employed to distribute the load between the various microservices and their instances. Also used to handle the SSL offloading of the external requests;
- **Elastic Compute Cloud (EC2) [105]** - Service utilized to run the microservices on the cloud;
- **Relational Database Service (RDS) [106]** - Service utilized to host the microservices relational databases;

- **Simple Storage Service (S3) [107]** - Service used to store static content.

4.2 C4 Model

The C4 model [6] represents the architecture of a software system in a hierarchical fashion. It is composed of four different levels, each one expanding on a system of a previous level, allowing the final user to first understand the architecture globally and then the details of each system that make it up. It is not mandatory to create every level of the C4 Model.

The four levels of the C4 model are:

- **Context** - Level one - The starting point of this model, it represents the "big picture", allowing the reader to see how the system interacts with users and other systems;
- **Container** - Level two - Follows the context diagram by expanding the system into different containers, showing its inner architecture and systems, such as databases and microservices;
- **Component** - Level three - This level expands each container of the previous diagram into components, allowing the reader to understand how each component interacts with the other;
- **Code** - Level four - The lowest level of the C4 model, the code level details each component in UML format.

The next subsections will present the diagrams related to the top three levels of the C4 model.

4.2.1 Context Diagram

Figure 4.1 represents the context diagram of Golden Goal's architecture. The two actors, the logged-off user and the logged-in user, interact with the system in different ways. The former can only see the website's content, such as reviews, experiences, and content related to football such as matches and teams. The logged-in user can interact with said content, marking matches as watched, and writing reviews, among other actions.

Golden Goal will get its information from two external APIs, API Football and Football-data API, as mentioned in Section 2.6.

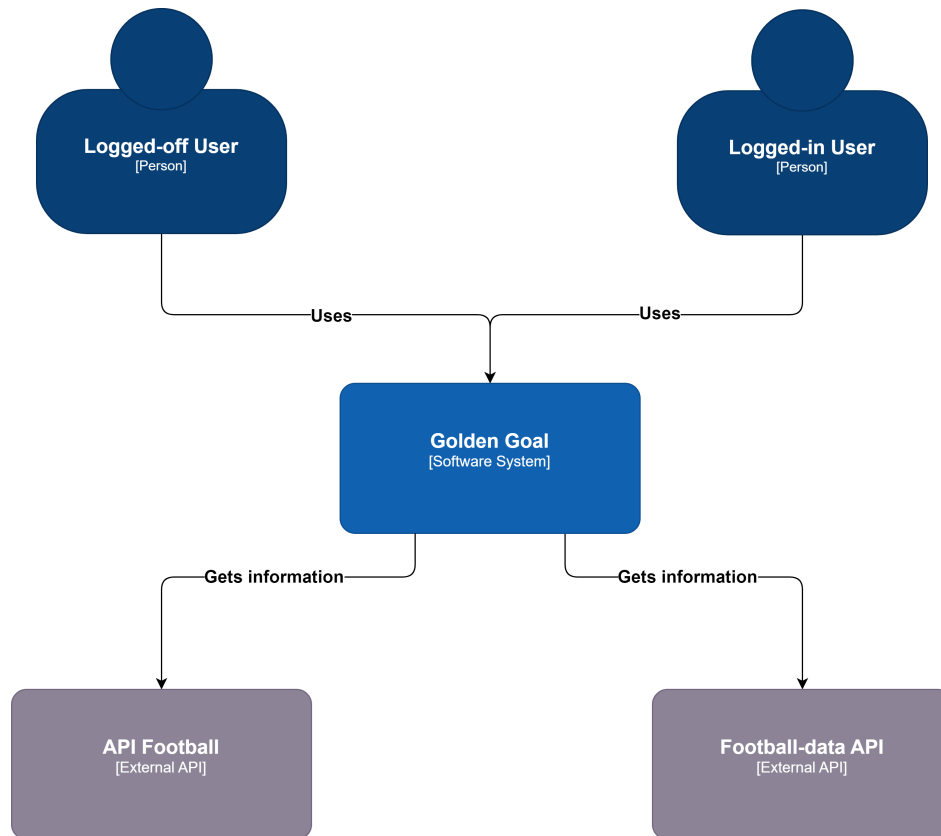


Figure 4.1: Context Diagram

4.2.2 Container Diagram

The container diagram in figure 4.2 gets to show the different components that make up the Golden Goal system seen in the previous diagram. The same actors continue to be seen, but the now expanded Golden Goal systems show how it is connected to the AWS services and external APIs.

The most important containers in this diagram are the three core microservices into which the application is divided, those being:

- Football Information;
- Content Manager;
- User Information.

Each microservice mentioned above is hosted in an EC2 instance and has a dedicated relational database using AWS's RDS service and PostgreSQL.

The Football Information microservice handles football-related data, for example, matches, competitions, teams, players, and managers. The specific information kept and used in this microservice is going to be discussed in Section 4.3. As this is the microservice that controls the football information, this is the one that is connected to both APIs chosen in Section 2.6. This microservice also communi-

cates with the S3 bucket, setting static content such as player photos and team crests.

The Content Manager microservice is responsible for handling the reviews, experiences, and comments. To accomplish this, it must interact with both the Football Information microservice and the User Information microservice to obtain information about the match associated with the review and the user that created the review, respectively. In a similar fashion to the Football Information microservice, it sets any static content the user adds, such as images, in the S3 bucket.

Finally, the User Information microservice keeps track of the user's personal data, such as his name, and his collections. For the collections, which can include either matches or reviews, this microservice needs to communicate with the Football Information or the Content Manager microservice, respectively.

As can be seen, the users interact with the system through a Single-Page Application (SPA), which is hosted on AWS's CloudFront. If the user is authenticated, the HTTPS requests sent by the SPA contain an Access Token, obtained by SPA's connection with Cognito. The requests then go through an external load balancer, created using Amazon's ELB, which does the SSL verification and offloading. If everything is valid, this load balancer sends the unencrypted HTTP/JSON request to the API Gateway. The SSL offloading improves the platform's performance, as requests sent to the backend are already unencrypted and don't need to repeat this process.

The API Gateway is a microservice without a database that acts as the single entry point of the backend, as talked about in Subsection 2.4.3. This contributes to the overall platform security, as this microservice will be able to filter all requests sent to the platform's backend. It checks for authentication and authorization using Amazon's Cognito and, if the requests are authenticated, redirects them to the internal load balancer.

This internal load balancer, which also uses Amazon's ELB service, redirects the request to the correct microservice instance. Considering that these microservices might be replicated to improve performance, the load balancer will improve the platform's availability and scalability, as it can check the health of the backend microservices and evenly distribute the load among healthy instances.

This architectural structure is very frequently used in the company.

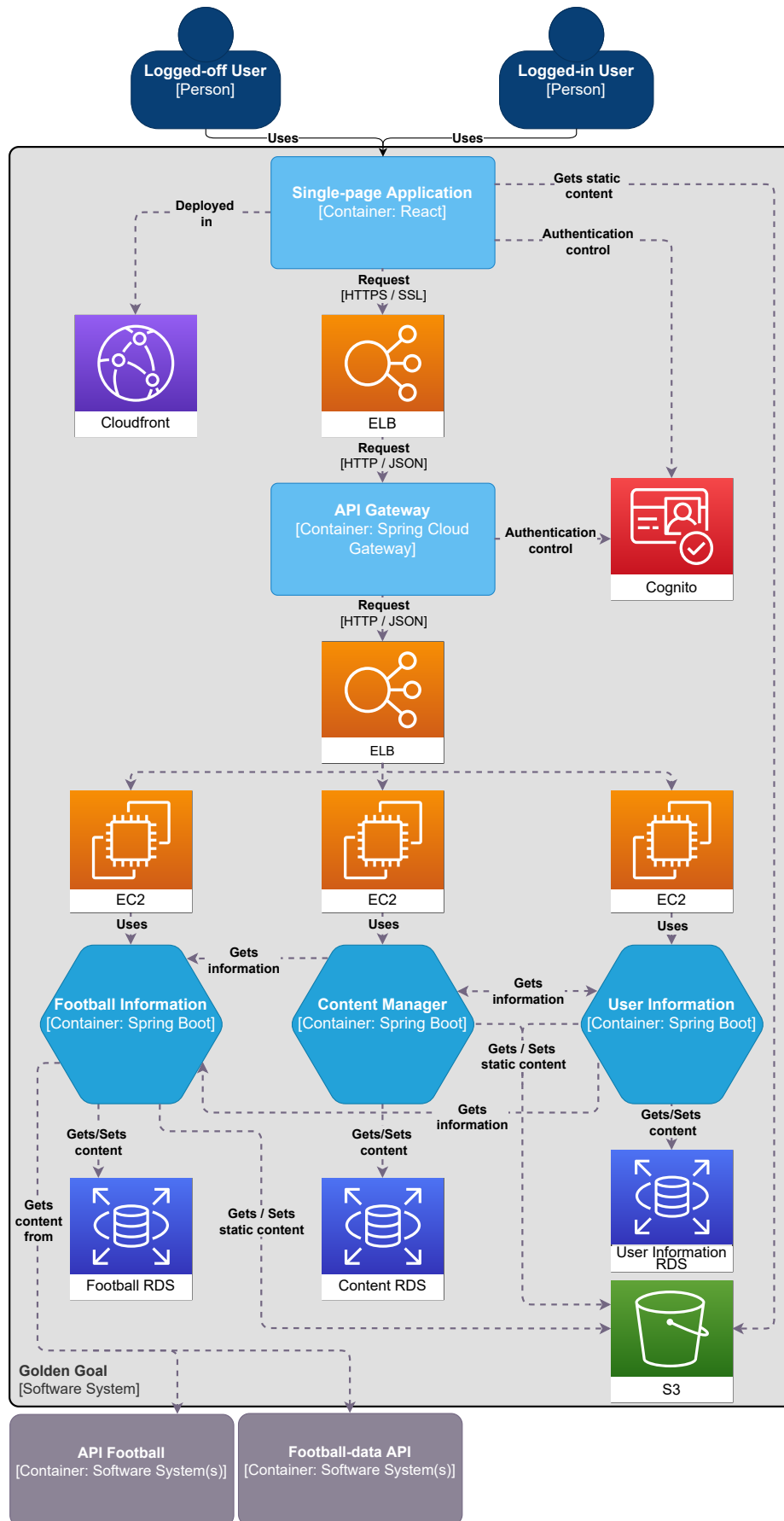


Figure 4.2: Container Diagram

4.2.3 Component Diagrams

The following diagrams are the lowest level of C4 diagrams created and break apart previous containers into each component they will have.

Single-Page Application

The SPA's component diagram, represented in figure 4.3, details the functioning of the frontend application. As it is going to be made in React, it features five views and five states, each one related to the requirements and user stories defined in Chapter 3. The navigation is made by the Router, which redirects requests to the views that keep any information needed in their state.

All views except the authentication view access the S3 bucket to get any static content needed, such as images. Other requests are sent through the API Gateway. The Session State gets the current authentication state of the user by communicating with Cognito.

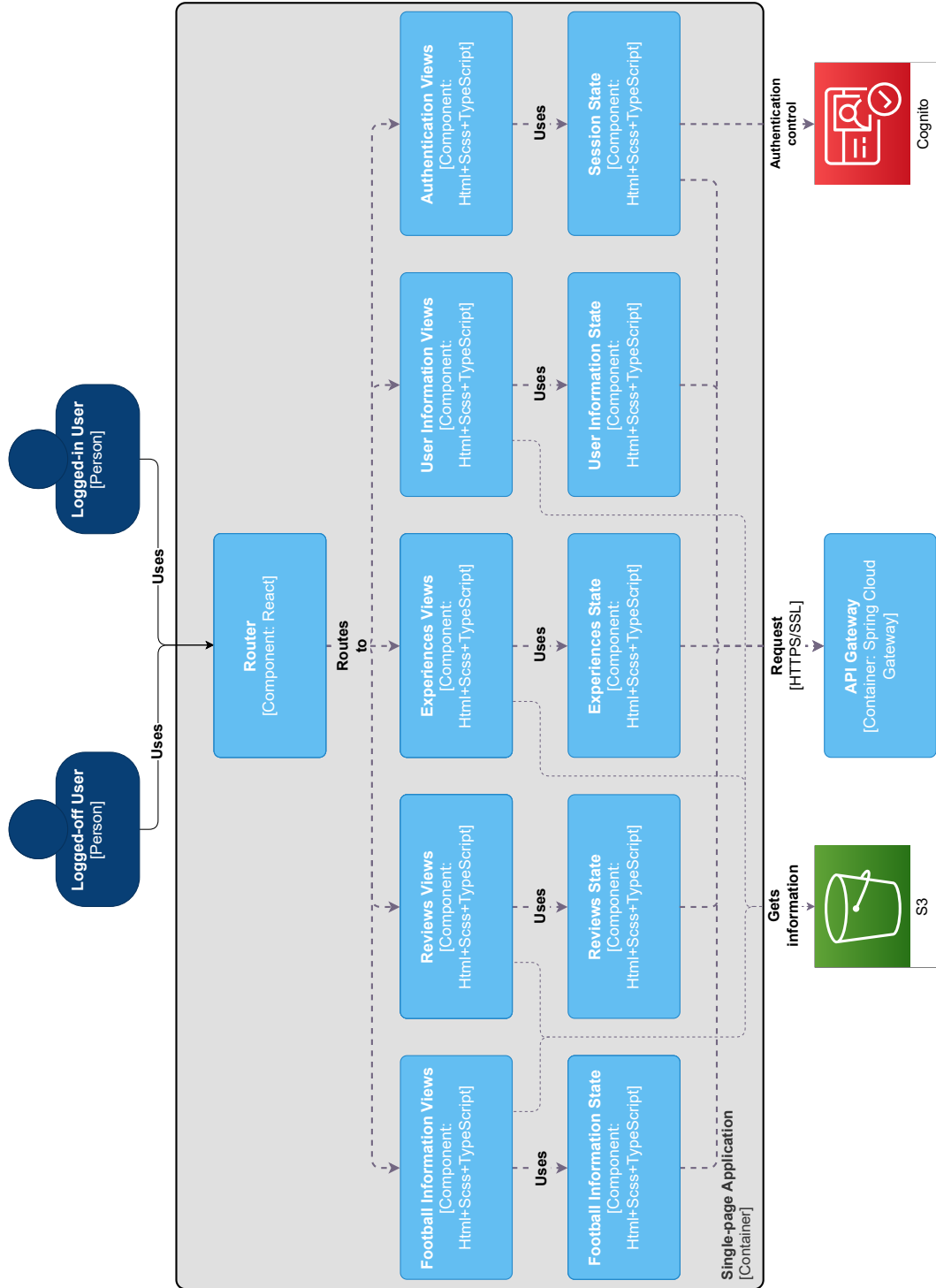


Figure 4.3: Single-page application component diagram

Football Information Microservice

The Football Information microservice component diagram, which is featured in figure 4.4, is split into five components - teams, matches, competitions, people, and statistics. The first four focus on general information, while the last one focuses on generating statistics. The statistics component only needs to access the information present in the relational database, while the four information components work with information present both in the S3 bucket, such as team crests, and the database.

Both APIs are accessed through a connector that must parse the information into our format, which will be seen in Section 4.3. General information can be gathered from either API, while statistics can only be gathered from the API Football, which means that the first four components are connected to both APIs while the statistics component is only connected to API Football.

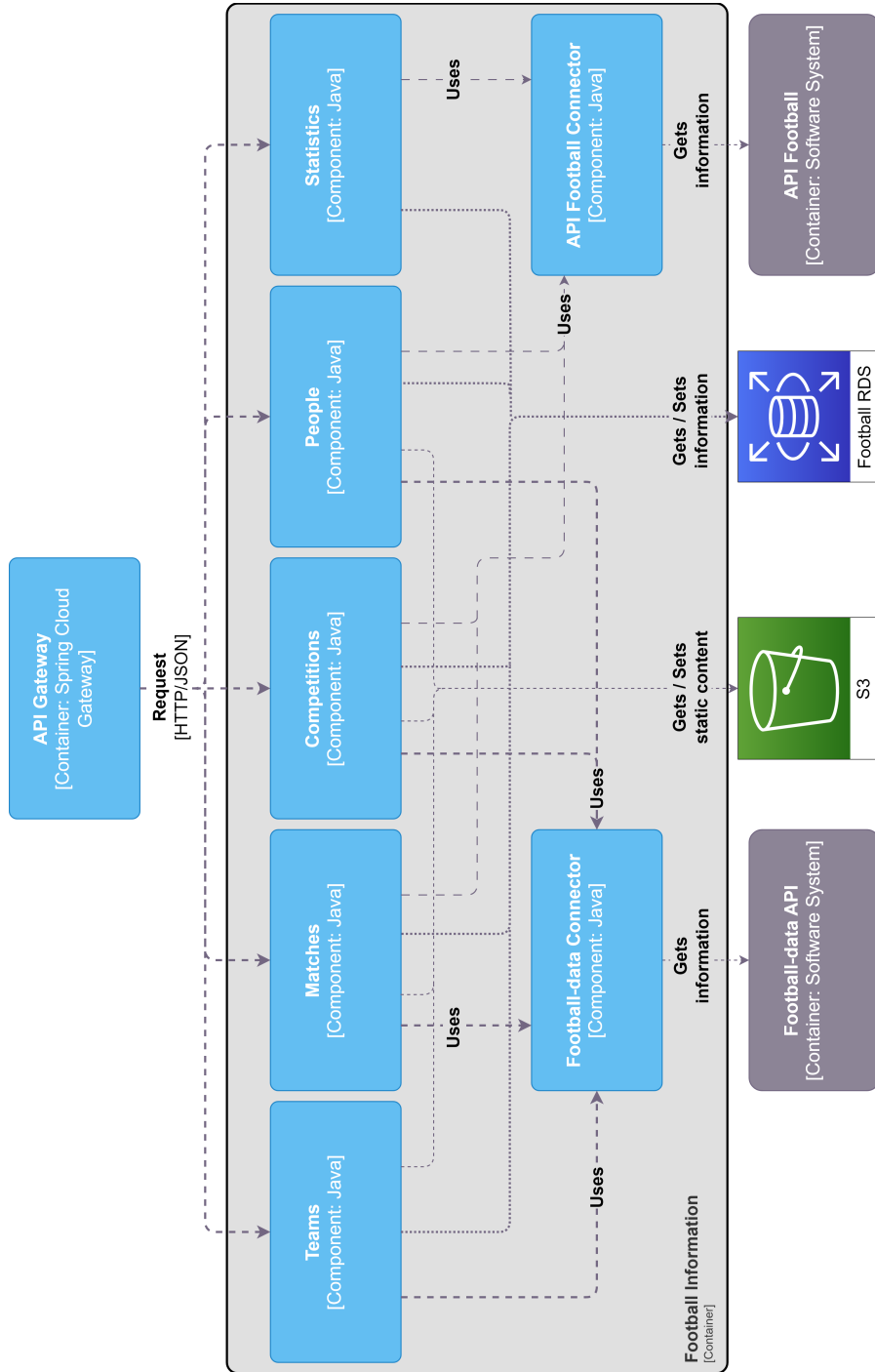


Figure 4.4: Football Information microservice component diagram

Content Manager Microservice

The Content Manager microservice component diagram, shown in figure 4.5 is split into the reviews component, the experiences component, and the statistics component.

The reviews and the experiences components are very similar, as both get information from the Football Information and User Information microservices to associate reviews and experiences with matches and authors. Their content is stored in the relational database as well as the S3 bucket, for static content. The statistics component computes review statistics for the author to access.

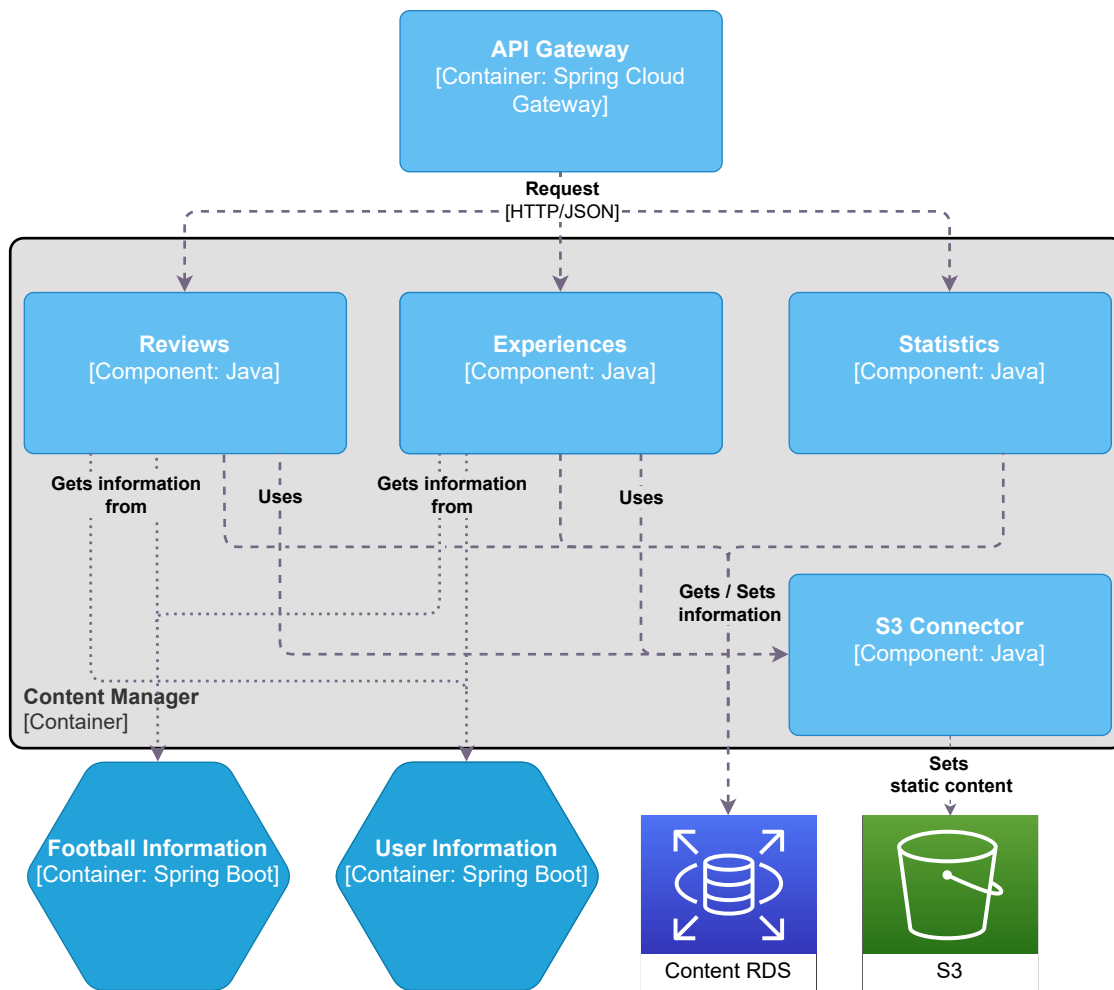


Figure 4.5: Content Manager microservice component diagram

User Information Microservice

This microservice’s component diagram is represented in figure 4.6. It shows three components: user information management, collection management, and statistics.

The first component manages the user’s personal information, such as name and profile image. Therefore, it needs to access both the relational database and the S3 bucket.

The second component manages the user’s collection lists, which can be lists of matches or reviews. Because of this, it needs to access the relational database and get content from the Content Manager microservice and the Football Information microservice.

Similarly, the statistics component, which computes user-specific statistics about reviews, needs to access the database and get information from the Content Manager microservice.

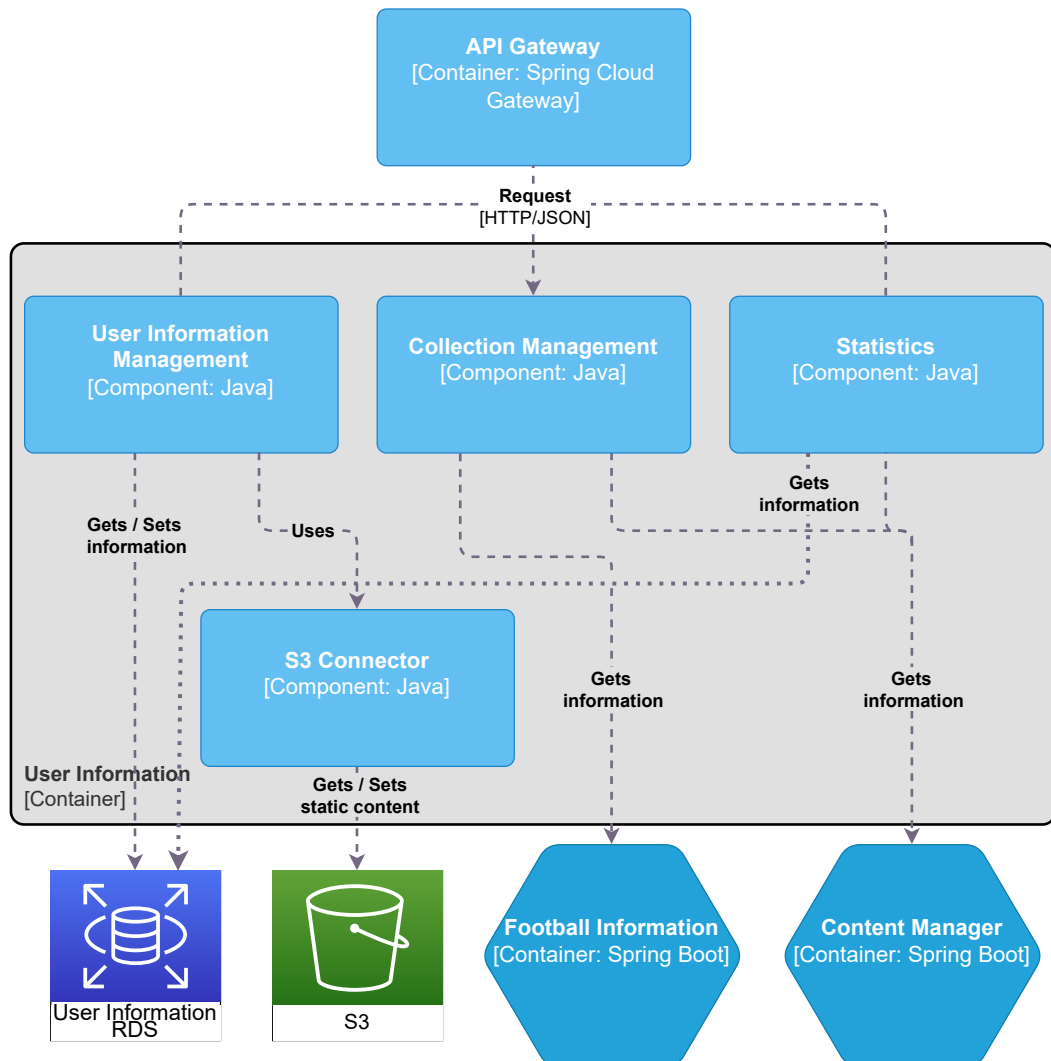


Figure 4.6: User Information microservice component diagram

API Gateway

This component diagram, as seen in figure 4.7, is divided into two components - Request Processing and Authentication Management. The first one works as a proxy, redirecting the authenticated requests to their respective microservice. The second one handles user authentication and session requests by using Amazon's Cognito, a service that contains a database featuring users' login credentials.

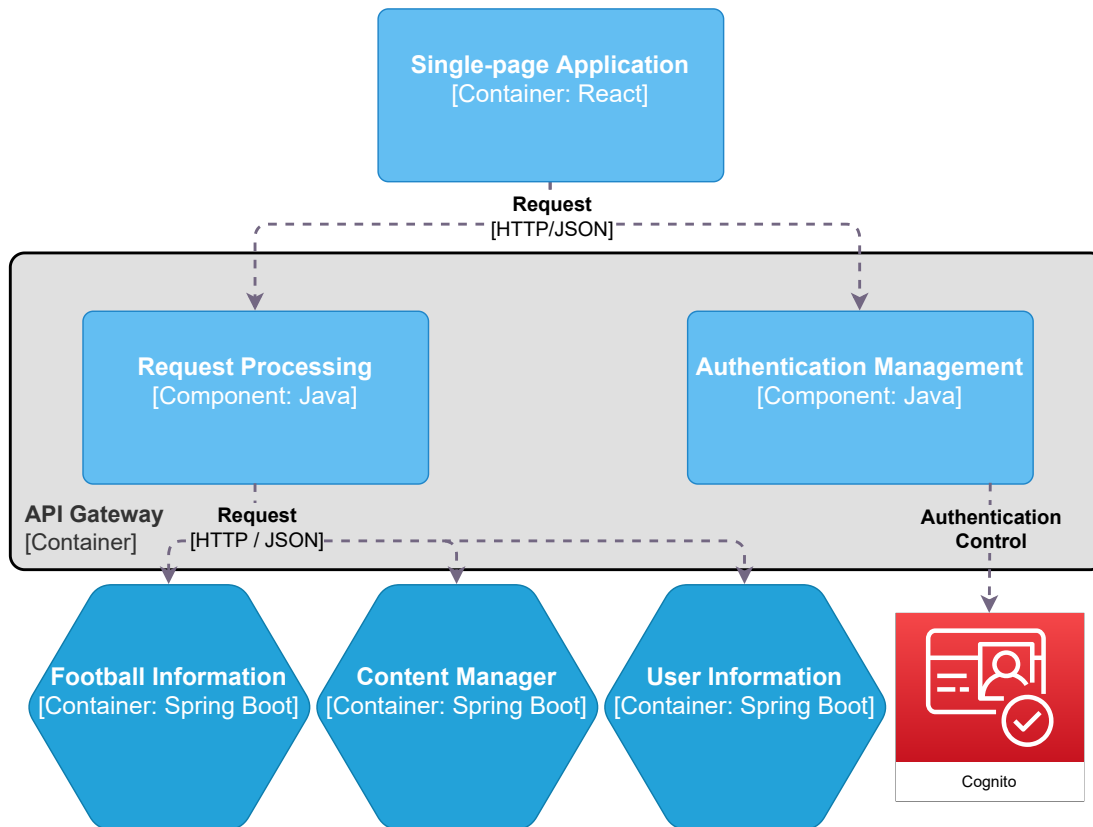


Figure 4.7: API Gateway component diagram

4.3 Entity–Relationship Diagrams

The Entity-Relationship Diagrams of a relational database help understanding which data is stored in which tables and how the tables are related.

As each microservice has its own database, three diagrams are going to be presented. In each diagram, tables are going to have the following fields, to help with database auditing:

- created_by and created_at;
- updated_by and updated_at;
- deleted_by and deleted_at.

However, to improve the diagram's readability, these will not be represented. All deletes made will be soft deletes, identified by the deleted fields, in order to avoid losing any data.

The tables in the diagrams will be colored to help distinguish foreign keys to external microservices' tables. The tables in blue belong to the Football Information microservice, the ones in green to the Content Manager microservice and the ones in orange to the User Information microservice.

4.3.1 Football Information Diagram

As this microservice handles all the football-related data, this diagram, as seen in figure 4.8, is inherently complex. It is best to start by understanding the core - the "teams" table and the "matches" table.

A team in the "teams" table can represent both a team and a national team, distinguished by the flag "is_national". Aside from the basic attributes, a team belongs to a country and has a stadium.

An element of the "matches" table represents a match between two teams, therefore having two foreign keys, one for each team, alongside the match's date, matchday (first, second, ...), stadium, and stage of the tournament (regular season, group stage, ...). To keep track of which managers participate in a match, the "matches" table has two foreign keys to the "managers" table, one for the home manager and one for the away manager.

However, the relationship to the players is different, as it is important to register the time at which a player is substituted in or out, as well as the position they are playing in the current match. To answer these needs, the "lineup_players" table was created, in which a player, with a position and both substitution times, is connected to a team and a match.

Coming back to the "matches" table, it is played in a competition, which is made up of various editions, for example, the 2022 world cup is an edition of the world cup competition. These editions belong to a season and have a winner, a member of the "teams" table.

There are also events, represented in the "events" table, associated with a match. They represent goals, fouls, and yellow cards, among others. These events are also connected to a team and a player, the creator of the event. Events have the time in which they occurred and are associated with a type, for example, a "yellow card". The "yellow card" type belongs to a category, which in this case would be "cards". A "red card" would also belong to the same category.

To finalize, it is important to understand how players and managers are stored, alongside their past teams. Managers' and players' primary key is their foreign key to the "people" table, allowing a person to be either or both. The "people_careers" table connects a person to their past teams by keeping a date for when the person was transferred in and one for when it was transferred out.

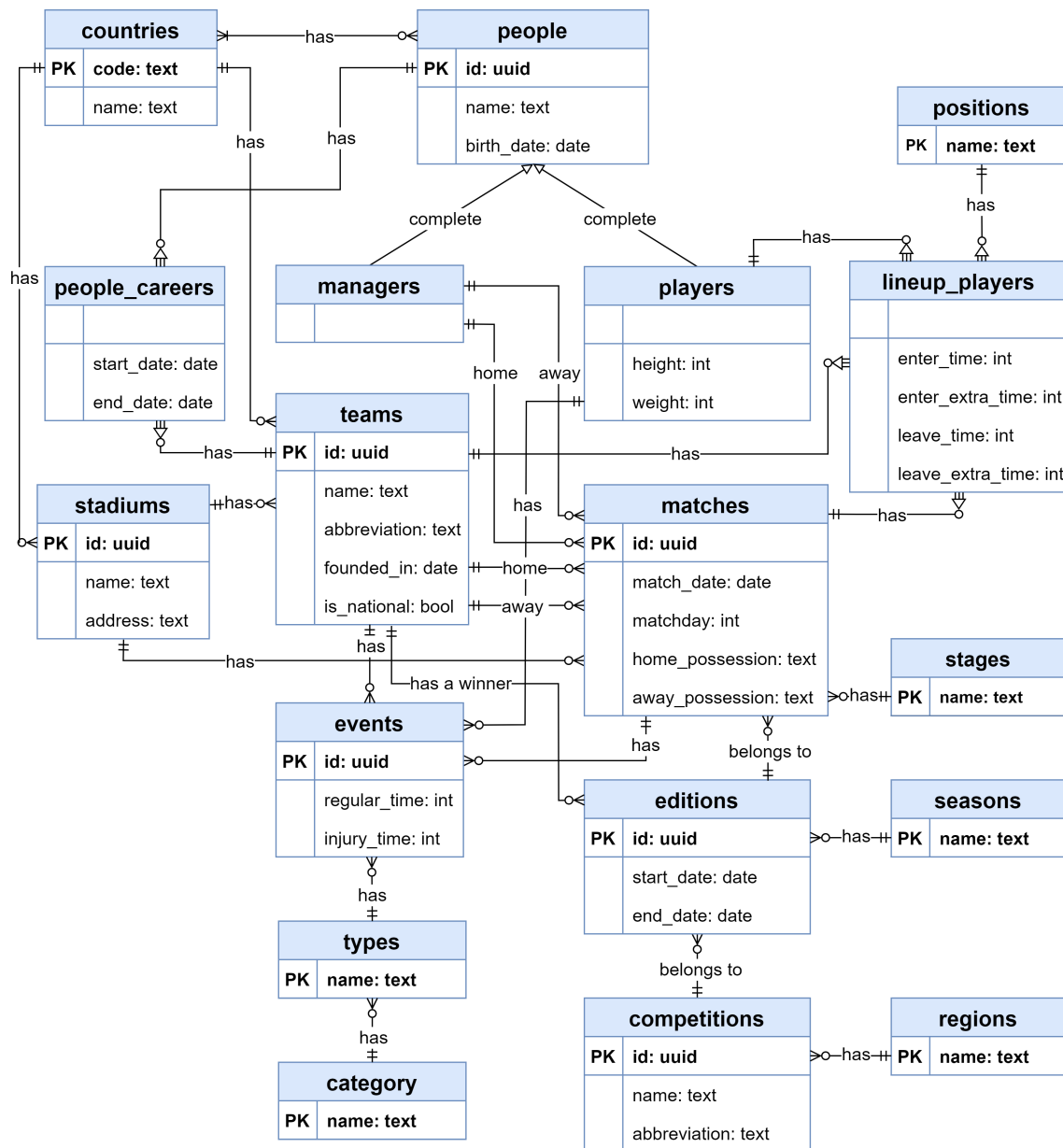


Figure 4.8: Football Information entity-relationship diagram

4.3.2 Content Manager Diagram

The Content Manager microservice diagram, as seen in figure 4.9, is much simpler than the previous one.

It is based on reviews and experiences, each of them with comments associated. As this database focuses on user-generated content, comments, reviews, and experiences have a foreign key to their author. Reviews and experiences also have a foreign key to the "matches" table, associating the content with a match.

As seen in Chapter 3, each review has its content level, tags, and language, while experiences only have the author's rating.

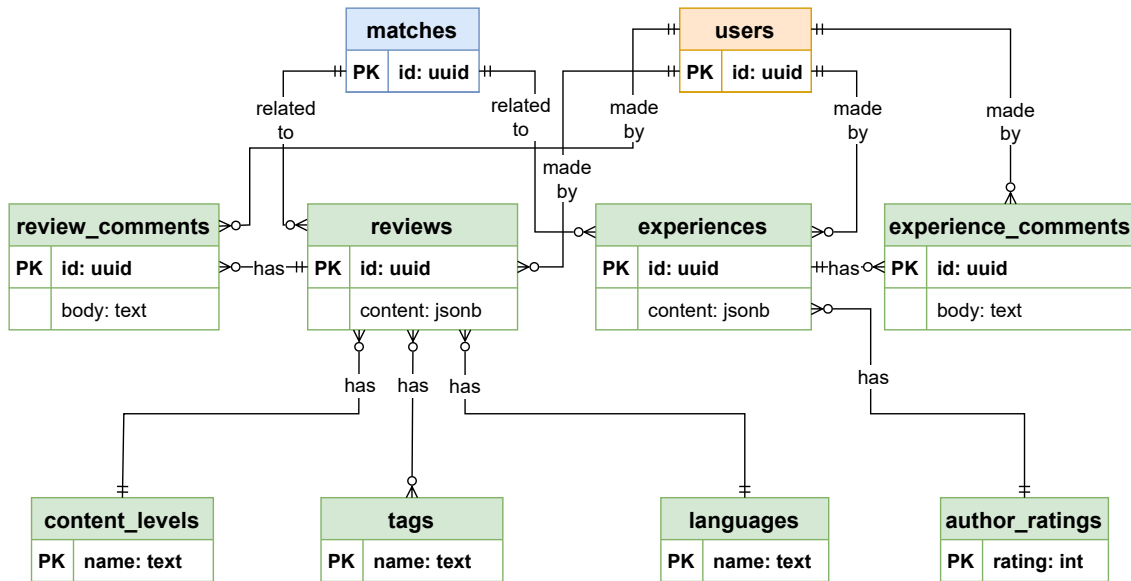


Figure 4.9: Content Manager entity-relationship diagram

4.3.3 User Information Diagram

This diagram, represented in figure 4.10, focuses on users and their content. Users can create comments, reviews, and experiences, as mentioned in the previous diagram’s discussion. Users can also follow teams, and, to create statistics, it is important to keep track of the reviews users have seen and shared. They can also mark matches as watched and liked.

Finally, users can create and manage collections, which can store matches and reviews. As matches watched, reviews liked, and matches liked are saved using default collections, they are not represented in the diagram.

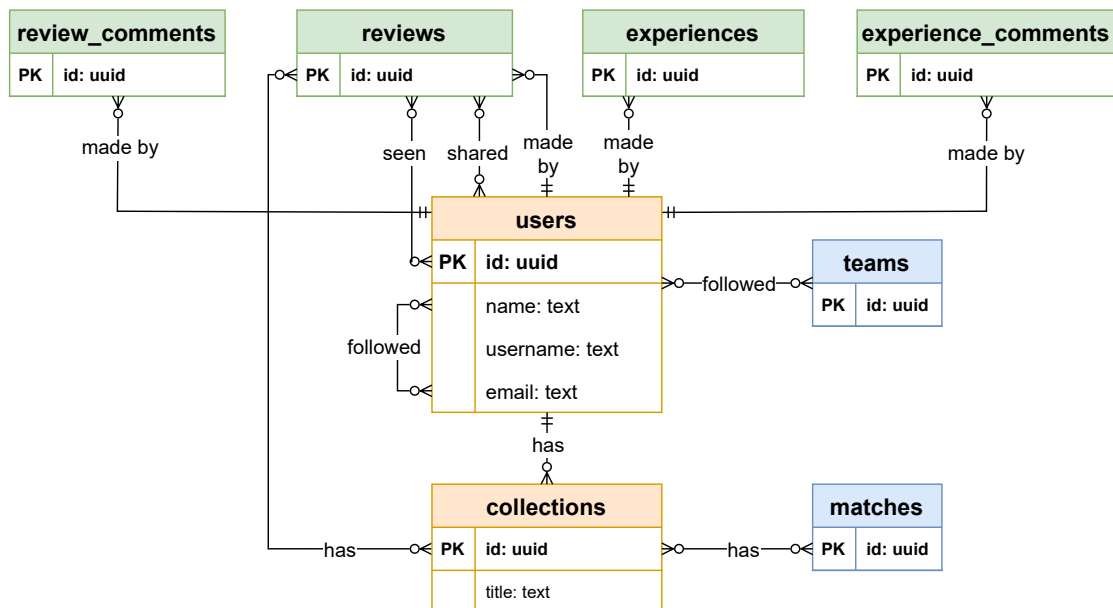


Figure 4.10: User Information entity-relationship diagram

Chapter 5

Methodology

Software development methodologies are the structured processes that are involved in the development of a project, from initiation to maintenance. There have been many methodologies used throughout the years, usually grouped into two categories - Traditional and Agile Methodologies [108].

The choice of a methodology is really important for a project, as it determines the way it is developed. It should consider not only the project itself but all the surrounding characteristics like client needs, objectives, and stability of the requirements.

In this chapter, both categories are going to be analyzed, comparing some of their advantages and disadvantages to find the one which best suits this project. This chapter will conclude with the chosen methodology, further diving into its processes, role distribution, and artifacts.

5.1 Traditional vs Agile Methodologies

Traditional Methodologies, also known as Sequential Methodologies, are based on clear processes that are phase-to-phase dependent on each other [108]. Each phase has a strict set of objectives and rules to follow, being necessary to complete these to proceed to the next phase of the plan.

Most Traditional Methodologies, like the Waterfall Model or the Unified Model, usually require an initial and long phase of planning, which is to be followed in all the next phases of development. This initial phase has an emphasis on prediction, needing to ensure that deviations from the plan are limited throughout the rest of the project.

Considering this, the methodologies stick to the following steps:

- Requirement definition;
- Platform design;

- Development;
- Testing;
- Deployment.

Traditional Methodologies tend to be more suited to predictable projects with clearly defined requirements, focusing on risk management and incremental development with detailed and heavy documentation.

As the software industry started growing, Traditional Methodologies started getting replaced with Agile Methodologies, the most popular being Scrum, Kanban, and Extreme Programming [109]. These are based on the Agile Manifesto [110] and focus on communication, flexibility, and planning.

Agile Methodologies tend to be more dynamic and people-oriented than Traditional ones, allowing them to adapt to changing requirements or conditions. That said, it is more suited to projects with more undefined requirements, allowing refactoring at a cheaper cost than Traditional Methodologies. Its smaller cycles allow more flexibility in deliveries of new features, adapting to the constantly updating projects created nowadays. It also allows for more proximity between the developing team and the clients with its focus on communication, which makes it more suited to smaller teams.

5.2 Chosen Methodology

The methodology chosen for this is going to be based on Scrum, a lightweight agile methodology that focuses on people and communication. It is composed of smaller cycles (Sprints), making it optimal for adapting to changing situations and requirements. It has been used in product development since the 1990s [111].

There are three different roles in a Scrum team [111]:

- The Product Owner (PO);
- The Development Team (DT);
- The Scrum Master (SM).

The first role, the PO, is responsible for maximizing the value of the product by managing and ordering the Product Backlog. This ensures that the work made in each cycle by the DT is the most valuable and beneficial to the progress of the project.

The DT is a self-organizing team that works and produces the items selected by the PO for each sprint, delivering an Increment of product at the end of each cycle.

Finally, the SM is responsible for ensuring that the Scrum is correctly implemented in the team. The SM makes sure that Scrum theory, practices, and rules

are being followed and that communication between members of the team is valuable and efficient.

The scrum methodology follows a sequence of events, as described in the following image:

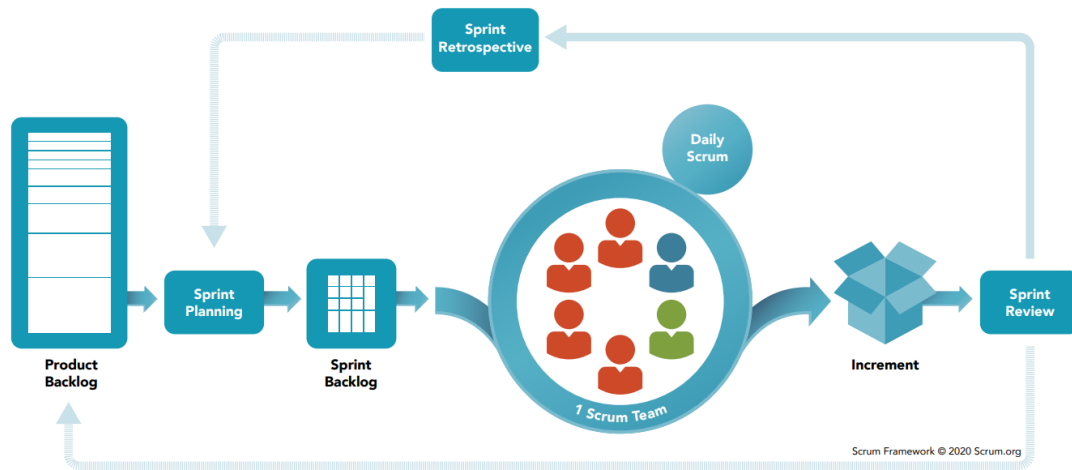


Figure 5.1: Scrum flowchart [112]

In this image can also be seen the next Scrum Artifacts:

- Product Backlog, an ordered list of everything that might be needed for the product. It is constantly evolving and ordered by the value each item brings to the project;
- Sprint Backlog, a set of items from the Product Backlog selected for a Sprint;
- Increment, all of the Sprint Backlog items completed during the Sprint.

The main event of this methodology, represented by the central elements of the picture, is the Sprint. This is a cycle of less than a month (usually two weeks) in which a Increment is produced. Each Sprint has a start and finish date, accompanied by a definition of what is to be built, a guiding plan for building it, and the final product - the Increment - to be delivered at the end of the Sprint.

The Sprint Planning, a meeting carried at the beginning of each Sprint, usually answers the two following questions: "What will be done in this Sprint?" and "How it will be done?". At the end of this meeting, a Sprint Backlog will be created containing all the items needed to satisfy the Sprint Goal.

There is a daily 15-minute meeting, called Daily Scrum, which is carried out by the DT to synchronize activities and make a plan for the next day. Usually in this meeting, each member of the DT explains what has been done since the last meeting, what will be done before the next meeting, and what obstacles are in the way of it.

At the end of each Sprint, there are two meetings, the Sprint Review and the Sprint Retrospective. The former focuses on identifying and demonstrating what

has been done and how problems were solved during the Sprint, discussing the product backlog and what to do next. The latter serves as an opportunity to inspect how the previous sprint went, identify what went well and what can be improved, and how to carry out those improvements.

As it can be seen, the Scrum methodology has a lot of advantages, being very adaptable and easy to understand. It is also very transparent, as everything that is classified as "Done" has to be approved by those performing the work and the final client. This contributes to client satisfaction, as he will be up to date with the progress of his product, and to team satisfaction, with this methodology's focus on communication and improvement.

As with everything, Scrum also presents some disadvantages. The client is involved in many stages of development which can slow it down, especially if the client is not available or repeatedly changes his ideas and, therefore, the requirements.

Considering both the advantages and disadvantages, it is easy to understand why this is the methodology chosen for this internship. Firstly, the disadvantages related to client involvement will not be a factor, as the client is the company itself and there should not be any communication issues. The transparency advantage will be important for the advisors to keep in touch with the student's progress and point out any problems as soon as they surface. The adaptability of this methodology will also be positive, as there will probably be a need for change as the student's experience with the technology evolves. Finally, the methodology is suited to smaller teams and is used by Grama, making this a better choice for integrate with the company and obtain quality feedback.

However, for this internship, Scrum will not be applied in its entirety. During the first semester, the product backlog will be defined, therefore, it will not be subjected to many changes during development. Also during the first semester, there will only be weekly Scrum meetings, contrasting with the second semester which will have daily ones. Considering that there will be only one person on the development team, all meetings will be shorter than usual, as there will not be the need to synchronize progress between the DT members. Finally, the planning, review, and retrospective of the Sprint will be stricter during the second semester, as the student's full focus will be on the internship, compared to a split focus between university and internship work during the first semester, which requires more flexibility.

Chapter 6

Planning and Risks

The next chapter addresses the planning of the project as well as the risks associated with it. Section 6.1 will address the project's planning, presenting Gantt diagrams [113] describing both the first and second semesters. Section 6.2 tackles the risk analyses made, one at the start, one at the end of the first semester, and one at the end of the internship, showcasing how the risks affected the progress of the internship.

6.1 Planning

This internship, which is scheduled for two semesters, is made up of two different stages, each one with the duration of a semester.

The first stage consists of the planning stage of the project, in which the student conducts relevant research and plans the entire project. It is in this stage that the student defines the project's requirements and architecture, as was seen in previous sections of this document.

As this stage was planned at the beginning of the internship, the next diagram describes both the initial order in blue bars and the order in which tasks were really executed, in green bars. The red bars represent tasks that ended after their planned time. The following diagram was made using TeamGantt's online Gantt diagram editor [114].

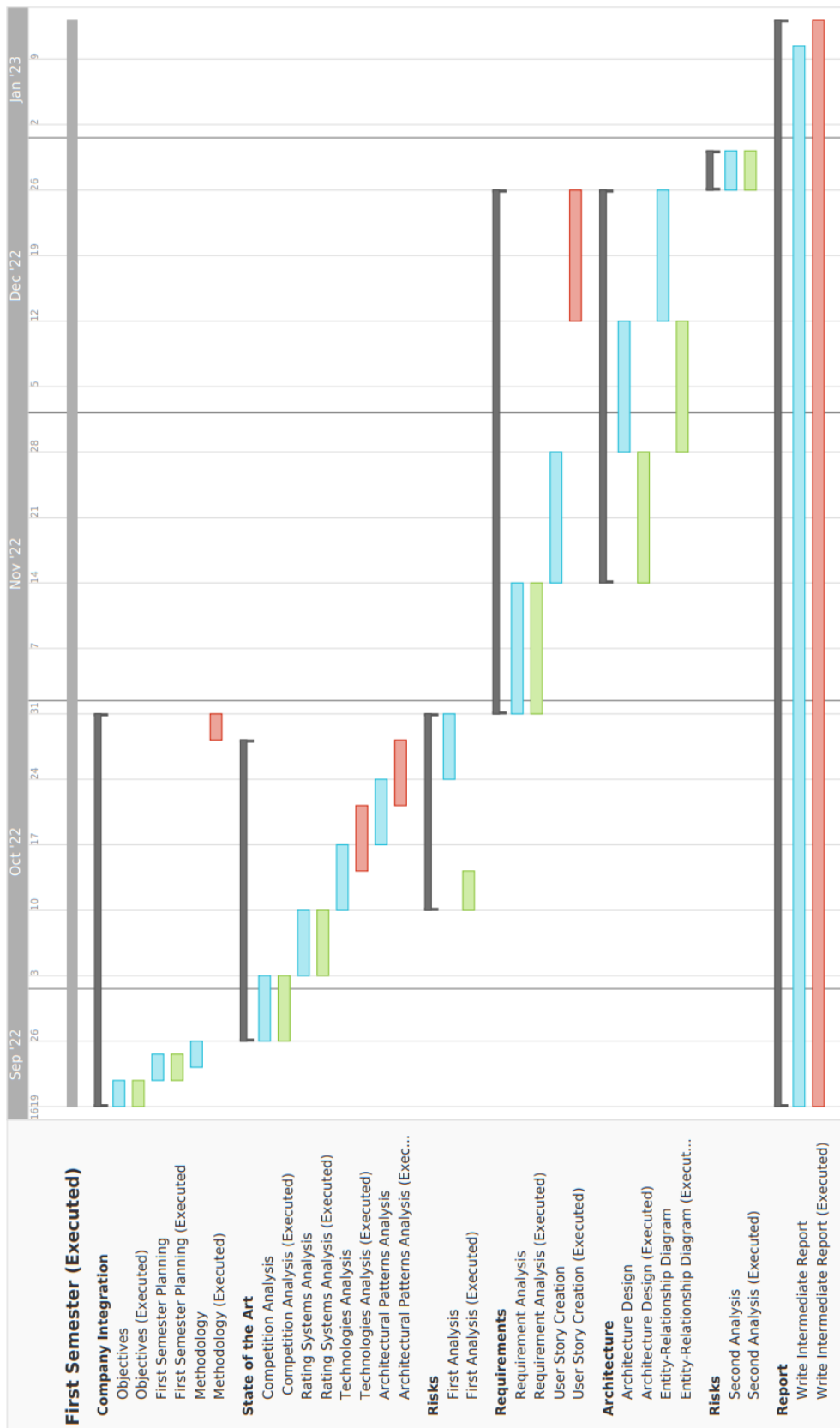


Figure 6.1: First semester

As can be seen by, even though there are a few shifts in the order, the overall structure of the planned diagram was correctly followed.

The initial idea was to start with the definition of the objectives, planning, and methodology. Out of these three, only the methodology research was slightly delayed, due to unexpected external circumstances which made it impossible to complete the task during the initial week.

During the following weeks, most of the work centered around the state of the art, focusing on analyses and research. As defining and understanding risks is crucial for the development of a project, the first risk analysis was anticipated. This allows for a greater contrast with the second risk analysis, which was conducted at the end of the semester. This anticipation caused the technology analysis and architectural pattern analysis to be completed after their estimated times, although none of the tasks exceeded their estimated duration.

Finally, the architecture was defined sooner than expected, allowing the company to finish defining the exact objectives and designs for the project. This made the user story creation finish after its estimated time, similarly to the analysis of technologies and patterns. The end of the project was stretched out by a few extra days in order to correctly finish this intermediate report.

The next figure represents the structure of the second semester in a similar structure to the previous diagram. It was based on the Scrum methodology and was composed of multiple sprints, with the last one dedicated to testing. Various ceremonies were realized during each sprint, although they are not represented due to a lack of space in the figure.

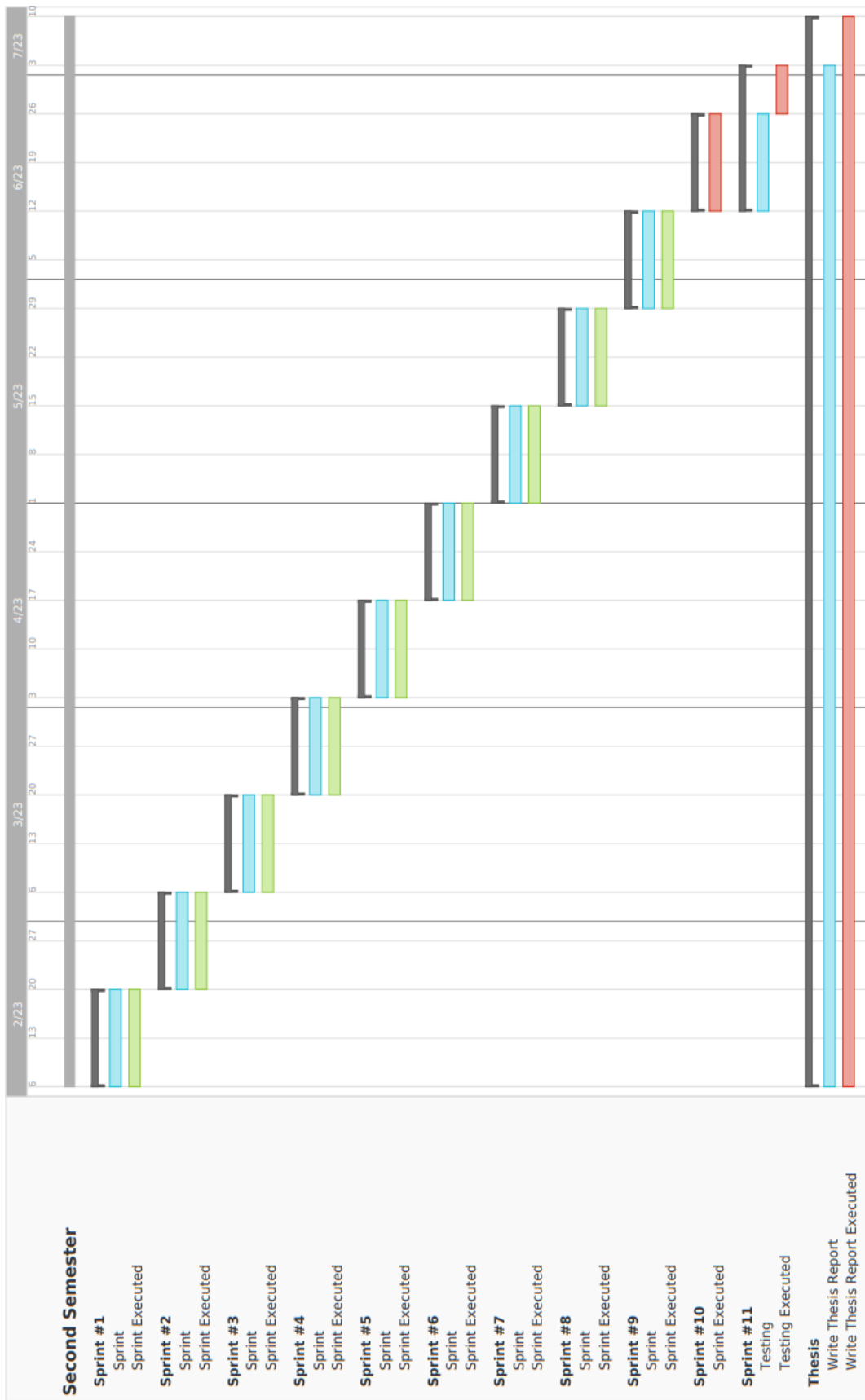


Figure 6.2: Second semester

Although only ten sprints were planned initially, there was a need to add another one before the testing sprint, in order to finish some of the functionalities that were in development. Considering this, the testing sprint was shortened and postponed two weeks and the last week of the semester, which was unoccupied, was dedicated to finishing the final thesis report that was written during the entire semester.

The order in which the requirements were developed will be discussed, with more context and detail, in Section 7.3.

6.2 Risk Analysis

Risks are inherent to all types of projects, regardless of team size, project scope, or project content. They are associated with uncertainty about the future and how it can impact the project's development [115]. As risks are uncertain, it is necessary to focus on everything that is controllable, that is, identifying these risks and assessing their impact, as well as stipulating how to handle them or how to make them less likely to occur.

As risk analysis must be ongoing to adapt to the progress of the project and to the ever-changing conditions surrounding it [116], two analyses are going to be presented, one made at the start of the semester and one made at the end of it, followed by a discussion about the changes that it underwent. This will be useful to understand how the progress made impacted the risks and will provide an opportunity to see if any of the risks impacted the semester and how effective the mitigation plans were.

6.2.1 First Analysis

In this section, all risks are going to be identified, along with their conditions, consequences, and mitigation plan, as well as their position in a risk matrix, according to their likelihood and severity. These risks are separated into two different categories - development risks and platform risks.

The first category will examine the risks inherent to the development of the platform, considering time constraints, inexperience, and the small team size. These will have a major impact on the success of the project, influencing the progress made throughout its duration. The platform risks will not be as important for the internship, as they will not be applicable to the platform until its release.

The second category will consider the risks associated with the characteristics of the platform itself, which is influenced by social media and football match-reviewing websites. These can change in the future, with further research and planning of the project, bringing up problems and possibly some solutions.

Development Risks

Id	Conditions	Consequences	Mitigation Plan
R-1	Poorly defined requirements	Development of unwanted or incorrect features. Non-fulfillment or delay of the objectives. Redefinition of the requirements	Intense planning and reviewing of the requirements. Redefining the project's objectives
R-2	Project is too ambitious	Non-fulfillment or delay of the objectives	Monitoring of the development. Redefining the project's objectives
R-3	Lack of experience with the chosen technologies	Non-fulfillment or delay of the objectives	Researching and comparing technologies. Studying the chosen technologies before the start of the development
R-4	Platform's UI design is not ready in time	Non-fulfillment or delay of the objectives	Monitoring of the development. Changing development order. Having a dedicated designer for the platform
R-5	Not enough time for university and internship work	Non-fulfillment or delay of the internship's objectives.	Scheduling time for university and the internship. Redefining the project's objectives

Table 6.1: Development risks - first analysis

These risks, unlike the platform risks, can have an impact on the overall progress of the project during the internship.

R-1 is a risk that is very interesting, as it can not only lead to development delays but to a poor final product as well. It is very important to fully plan and understand the requirements before developing the final product, as creating unnecessary or incorrect features has a massive impact on the scheduling. Similarly to this, the redefinition of the requirements, in the case they were incorrect or incomplete, also delays the project's progress.

R-2 can occur either if the scope of the project is too large or if the duration of

the tasks is incorrectly estimated. The first cause has a really low probability of happening, as estimations before the development will lead to a prioritization of the tasks, controlling the overall scope of the project. This risk becomes more prevalent with the second cause, as incorrect estimation may be likely to happen due to inexperience with the technologies and, therefore, inexperience with the amount of time needed to complete each task.

The R-3 risk can affect the estimation of task duration, which is very likely to happen, considering the variety of technologies to be used and the limited experience with them. While the likelihood of occurrence cannot be changed, further research into the technologies can lessen this risk's severity.

R-4 is very unlikely to happen, as there are designers from the company assigned to this task. Also, its severity is really low, considering that the development can start with the already existing UI or backend tasks.

Finally, R-5 is relatively likely to happen, as there is an overlap between the university and the internship in this first semester. It is important to mitigate its impact by carefully planning and dividing the workload correctly.

Platform Risks

Id	Conditions	Consequences	Mitigation Plan
R-6	The user base does not grow past its initial state	Less content and user interactions on the platform. User abandonment	Promoting the platform in popular football-oriented media platforms (online, television, ...)
R-7	Excessive toxicity in the platform	Less user interaction and posts. User abandonment	Increasing moderation. Adding a report function. Researching rating systems to promote good behavior
R-8	Uninteresting or irrelevant content promoted	User abandonment	Researching rating systems to ensure the quality of ratings. Better content promotion algorithms

Table 6.2: Platform risks - first analysis

As mentioned previously, these risks do not impact the platform before its release. However, it is important to develop their mitigation strategies in advance, so that it is easier to implement them when they are needed.

R-6 is the most critical of all these risks, considering that without a user base it is impossible to have a social platform. Promotion of this website will be crucial and

should be one of the first steps after the release, regardless of the media platform chosen to promote it.

R-7 and R-8 are interesting, as they are common with a lot of social media platforms. The first one is augmented by football and all its controversies, and it has to be dealt with. Increased moderation with a report function is a way to deal with the toxicity, while better rating systems are a way to prevent it. The second one will quickly lead to users leaving the platform, therefore good ratings and content promotion algorithms will be crucial for the website’s longevity.

Risk Matrix

The risks mentioned above are presented by their Id in 3x3 a risk matrix [117], where the horizontal axis represents the impact of each risk and the vertical axis means the likelihood of the said risk happening.

The values of the vertical axis are divided into 3 different categories, each with its own likelihood of occurrence, as can be seen in the following table:

Category	Likelihood
Low	0 - 39
Medium	40 - 69
High	70 - 99

Table 6.3: Likelihood categories

The impact axis is also divided into 3 categories, each described in the following way:

Category	Impact description
Low	The consequences are easily manageable
Medium	The consequences are manageable with great effort
High	The consequences are extremely hard to recover from

Table 6.4: Impact categories

Putting all of this together, the following risk matrix is obtained:

Likelihood	High	Medium	High 3 7	Critical 6
	Medium	Low	Medium 5	High 8
	Low	Low 4	Low 2	Medium 1
		Low	Medium	High
		Impact		

Figure 6.3: Risk Matrix - First Analysis

The R-6 platform risk has the highest severity of them all. This is because a social platform cannot survive without a user base, the core of all content and interaction on the website. Therefore, this is a high-impact risk for the platform. Due to the heavy competition on the social media and football website areas, its likelihood of occurrence is equally high, making this the only critical risk for the platform.

It can be seen that risks R-1, R-5, and R-8 are also included in the highest impact column. The first two can cause the project to fall way behind schedule, whether by leading to unwanted features or by not balancing the time correctly. Even though they have the same impact, R-1 is less likely to happen because of the intense review of the requirements. The latter is a platform risk that can diminish the user base, even if it is not as drastic as R-6. For that reason, it is deserving of this position on the impact scale.

The three risks with moderate impact are R-2, R-3, and R-7. R-2 can have an average impact on the project but is not as likely to happen as some other risks due to constant monitoring, planning, and reviewing of the requirements. R-3 is one of the most likely risks enumerated, considering that it is almost certain that the unfamiliarity with the technologies will affect the development efficiency. Its impact, however, can be reduced with further exploration of the technologies used. Finally, R-7 affects the user base, similarly to R-6 and R-8, but has a lower impact than those, as toxicity is not the main factor that leads to a declining user base.

The lowest impact risk, R-4, has really low repercussions, as the order of development can change to accommodate problems with the design’s progress. It is also very unlikely to happen, due to the platform having a dedicated designer.

6.2.2 Second Analysis

The structure of this subsection will be similar to the previous one, starting with the development risks, followed by the platform risks, and ending with the placement of each one in the risk matrix.

However, as the previous section focused on introducing and explaining each risk and decision made, this will focus on the differences between the two and what caused those differences.

Development Risks

Many of the risks identified in this analysis are the same as the ones identified in the previous analysis, even though their placement on the risk matrix will be different for some.

The highlight of this subsection is the removal of risk R-5, related to the split time between university and internship work. As this analysis is being carried out right before the second semester, the student's total focus will be on the internship, as there will not be more university work. Therefore, R-5 has become a non-risk and will not be featured further.

Considering this, the development risks going into the second phase of the internship are the following:

Id	Conditions	Consequences	Mitigation Plan
R-1	Poorly defined requirements	Development of unwanted or incorrect features. Non-fulfillment or delay of the objectives. Redefinition of the requirements	Intense planning and reviewing of the requirements. Redefining the project's objectives
R-2	Project is too ambitious	Non-fulfillment or delay of the objectives	Monitoring of the development. Redefining the project's objectives
R-3	Lack of experience with the chosen technologies	Non-fulfillment or delay of the objectives	Researching and comparing technologies. Studying the chosen technologies before the start of the development
R-4	Platform's UI design is not ready in time	Non-fulfillment or delay of the objectives	Monitoring of the development. Changing development order. Having a dedicated designer for the platform

Table 6.5: Development risks - second analysis

Platform Risks

This section does not have any changes, as these risks are about the future and sustainability of the platform after its release and will not affect it before that. Considering this, there still are some changes related to the placement of these risks on the risk matrix, as further research has been done during the semester.

The platform risks going into the second phase of the internship are the following:

Id	Conditions	Consequences	Mitigation Plan
R-6	The user base does not grow past its initial state	Less content and user interactions on the platform. User abandonment	Promoting the platform in popular football-oriented media platforms (online and TV)
R-7	Excessive toxicity in the platform	Less user interaction and posts. User abandonment	Increasing moderation. Adding a report function. Researching rating systems to promote good behavior
R-8	Uninteresting or irrelevant content promoted	User abandonment	Researching rating systems to ensure the quality of ratings. Better content promotion algorithms

Table 6.6: Platform risks - second analysis

Risk Matrix

For this section the risk matrix used is the same as in the previous analysis, following the same categories and parameters for each axis.

Bearing that in mind, these are the new risks placed on the risk matrix:

Likelihood	High	Medium 3	High	Critical 6
	Medium	Low	Medium 7	High 8
	Low	Low 4	Low 2	Medium 1
		Low	Medium	High
		Impact		

Figure 6.4: Risk Matrix - Second Analysis

There were two risks whose placement changed in the matrix, R-3 and R-7. The former has decreased in impact, as the frontend and backend technologies have been researched and both the student and members of the company have experience with the chosen technologies. This has brought R-3 down a level in the impact scale. Its likelihood has not changed much, as this is a risk that would happen independently of the chosen technologies due to the student's own inexperience in projects of this scale.

R-7 has been brought down a level in the likelihood scale. The research done on the rating systems influences user behavior enough to deter review bombing, one of the main concerning factors of the toxicity that is inherent to social websites. The toxicity related to the inappropriate and offensive comments is not going to be addressed in this Minimum Viable Product (MVP), as it needs a relatively large user base to become a problem. The impact of toxicity itself has not changed, making this risk remain in the same column.

The risks R-1, R-2, and R-4 became less likely with the progress made during the semester, even if the limited granularity of this matrix will not allow for changes in their likelihood level. This is especially true for R-4, as the platform's UI design is almost complete and its likelihood is nearing 0%. R-8's likelihood has decreased but not enough to bring it down to the "Low" level. The research about the rating systems will help in fighting this risk, but this risk will only be truly mitigated when there is a content promotion algorithm that adapts to each user, something that is out of the scope of this internship and irrelevant for an MVP.

R-6 suffered almost no changes. Whilst it is important to consider this risk, working on the promotion of the platform itself is not a part of this internship. Therefore, there were no changes related to this risk.

6.2.3 Final Analysis

This analysis, conducted post-development, aims to assess the occurrence of risks and evaluate the effectiveness of the mitigation plans. It focuses specifically on the Development Risks, as the Platform Risks pertain to the future of the platform after its release and were not relevant during the development phase.

R-1: Poorly defined requirements

This risk did not manifest itself during development. While it could have had a large impact on the development, the mitigation plan of "intense planning and reviewing of the requirements" significantly minimized the probability of it occurring.

This result is aligned with the positioning of this risk on the Second Risk Matrix, showcased in Figure 6.4.

R-2: Project is too ambitious

R-2 was one of the risks that occurred during development. The time restrictions of the internship, in conjunction with the speed of development, made the fulfillment of all the original "Must have" requirements impossible.

However, this risk did not severely impact the development, as it was swiftly mitigated by the scope redefinition that will be detailed in Section 7.2. This careful analysis found requirements that were being over prioritized and did not have as high of an importance as their priority would suggest. As they were not crucial for the final MVP, their removal did not affect the success of the internship.

Reviewing this risk's positioning on the Second Risk Matrix, while the impact is correctly placed in the middle column, the likelihood of it happening should have been higher, as the student was entering a new environment, which had higher quality standards and processes than his previous experiences.

R-3: Lack of experience with the chosen technologies

The R-3 risk was the other risk that impacted development. While this risk's impact was mitigated by the chosen technologies, on which the student had some experience, the student had never studied them this carefully and used them to such depth.

This risk was particularly challenging to mitigate in some cases, such as the rich text editor library, among others. The text editor had a great degree of uncertainty, even after the research done to in the first semester of the internship, as it had never been used by the student or anyone in the company.

While this is a negative factor to the speed of development, which in turn contributed to the occurrence of the R-2 risk, it is also a positive aspect for the student, which had to improve and develop new skills to surpass the challenges faced.

This risk was correctly placed in the risk matrix's likelihood, as it was extremely hard to mitigate its occurrence and was deserving of the highest level in this measure. However, its impact should have been in the "Medium" column, as the technologies, although known and used previously, had never been used to this extent and with such quality standards.

R-4: Platform's UI design is not ready in time

This risk did not materialize during the development, as it was correctly mitigated by the presence of a designer dedicated to the platform, which made all the adjustments needed prior to the start of the development.

Therefore, it was correctly placed on the risk matrix, as it would have had a small impact and the mitigation plan was very effective, as explained in Section 6.2.2.

Chapter 7

Development

This chapter provides an overview of the development process of the Golden Goal platform, during the second semester of the internship. It begins with Section 7.1, which describes the development team and task organization.

Following that, Section 7.2 discusses the redefinition of the project's scope, which was necessary to align the objectives of the internship to the limited time constraint. Afterwards, Section 7.3 presents the project's timeline of development.

Sections 7.4 and 7.5 list the functional requirements and the architecture that were developed, respectively. Lastly, Section 7.6 is used to showcase the platform's main developed functionalities.

7.1 Development Process

This section focuses on describing the development process of the Golden Goal platform. Firstly, the process organization is described in Subsection 7.1.1. Afterwards, Subsection 7.1.2 describes in detail how the tasks were organized.

7.1.1 Process Organization

The team associated with the Golden Goal project is made up of the development team, comprised solely of the student, the design team, the code review team, the scrum master, and the product owner. Additionally, the internship's advisor from the company served both as a member of the code review team and as the scrum master.

The first step of the process was the creation of the platform's mockups, which were developed prior to the internship's beginning. These were improved during the first semester, following the meeting with the football analyst, mentioned in Section 3.2.

Subsequently, the development part of the internship commenced. In this phase

the student implemented the platform's functionalities according to the user stories and functional requirements, as defined in Sections 3.2 and 3.3, respectively. The newly developed functionalities were subject to code reviews, conducted by the members of the code review team.

Throughout the development process, unit tests were created to ensure the proper functioning of the implemented functionalities. Further testing was conducted during the last sprint of this internship, focusing not only on the functionalities developed but also on the fulfillment of non-functional requirements.

As the adopted methodology is based on Scrum, the ceremonies - daily meetings, sprint plannings, sprint reviews and sprint retrospectives - were attended by the student and members of all participating teams.

7.1.2 Task Organization

The tool used to plan and structure the tasks for each sprint was Linear [118]. It resembles a Kanban board and was structured along seven different columns:

- **Backlog** - all the tasks to be developed;
- **To do** - tasks to be developed during the current sprint;
- **In Progress** - tasks in development;
- **In Review** - developed tasks in code review;
- **Ready to Test** - finished tasks that are ready to be tested;
- **Done** - finished tasks that passed the tests;
- **Cancelled** - tasks that were not implemented and were cancelled.

During the sprint refinement ceremony, each task was assigned an effort value. These values were based on the Fibonacci sequence (1, 2, 3, 5, 8) and represented the task's complexity. Higher values correspond to complex and long tasks, while lower values vice-versa. These values were used to plan each sprint's workload and were decided by the members participating in the ceremony.

Figure 7.1 shows an example of the Linear board. The tasks are distributed among the columns and each task has its own name, priority, and complexity. Tasks can then be expanded to read more details about them.

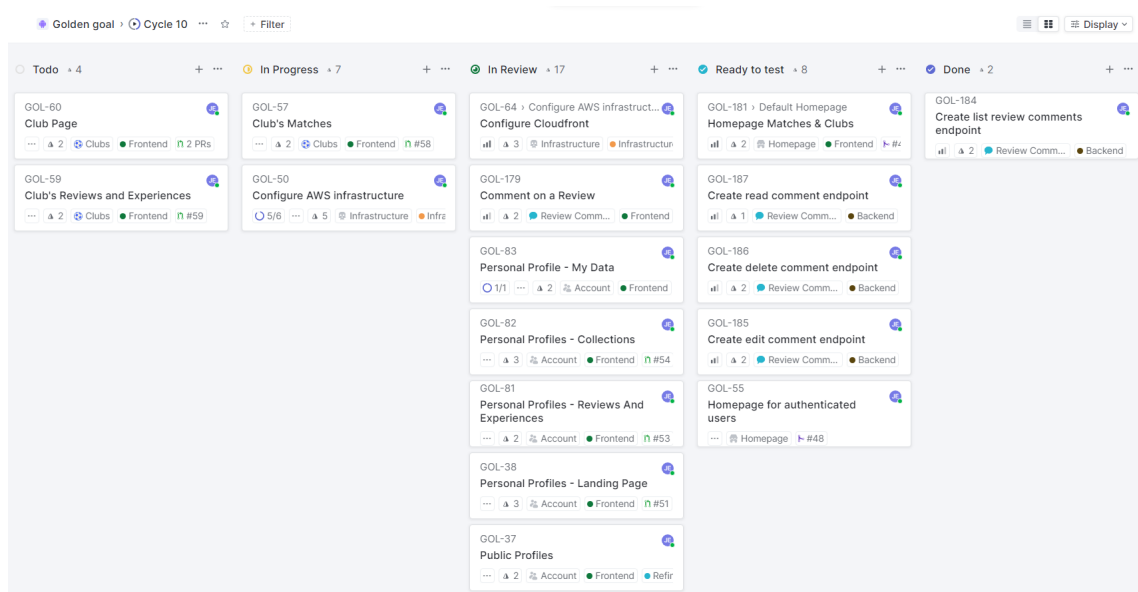


Figure 7.1: Linear sprint cycle board

7.2 Scope Redefinition

Upon commencing the development, it became evident that the scope was too extensive for the duration of the internship. This was due to the student's limited experience in working on larger projects and with the selected technologies. Consequently, a requirement and architecture analysis was conducted, taking advantage of the agile methodology chosen, to identify potential targets that could be deferred for future work.

The first target identified was the implementation of Experiences. These were deemed to be quite similar to Reviews in terms of technologies and logic involved, and would require some time to complete. Additionally, in the context of the final product, the Experiences were not the focus of the MVP. Its primary objectives were to develop a football website which allowed users to navigate through Competitions and Matches, as well as creating a text editor that allowed users to compose Reviews with diverse content.

The second target was the player and statistics component of the football information microservice. Retrieving this information would necessitate integrating another API, which would require substantial setup time. Furthermore, this API imposed a monthly fee, as the limited number of daily requests allowed in its free tier was insufficient to obtain information on such a large number of Teams and Matches, even if these were restricted to a single competition. By deferring these components, only one API would need to be used to retrieve all of the football information necessary, avoiding the need to research and set up another API. As is the case with the Experiences, this work would also resemble something previously developed, and would therefore bring diminished value to the internship. Moreover, the players and the statistics would only be minor details in this MVP,

which is focused on the Competitions, Matches, and Review creation.

Thirdly, although previous seasons of Competitions could be obtained from the already implemented API, they were assigned a low priority. They were deemed non-essential for the MVP, as they would require significant implementation and would need many requests to the API without greatly enhancing the core functionalities of the MVP.

Finally, the decision was made to remove the internal load balancer from the scope of the internship. This was because the scalability and availability it provides are not crucial for the MVP. Additionally, it would require a monthly fee to AWS, as load balancers cannot be disabled and the free tier only supports one load balancer per month, and setting it up would be similar to the external load balancer.

7.3 Development Timeline

Linear also provides a timeline of task development across different categories, as can be seen in Figure 7.2.

The project's infrastructure spanned the entire development process. It included tasks from creating a project, which were completed in the beginning, to deploying these projects in the AWS, one of the last tasks completed.

The first category to be completed was the style guide, which was fully developed during the first sprint and served as the foundation of the frontend application.

During the next five sprints, the focus shifted to setting up the applications authentication with AWS's Cognito, alongside creating the User Information project and necessary endpoints. Simultaneously, the Football Information project was also started, which involved researching about the Football-Data API, gathering data from it, and setting up all endpoints to use in the future. Additionally, basic frontend components were also created, such as buttons, checkboxes, and drop-down menus, among others.

Sprint number seven, which started on the first of May, revolved around creating reviews. The necessary backend was developed, alongside the creation of the "Add review" page, which was implemented using the Lexical rich text editor.

The following sprint focused on finishing the development of the review pages and creating endpoints related to Collections and User statistics and interactions, such as liking a match. These would be crucial for the last two sprints.

Lastly, the final sprints were focused on creating the remaining pages, such as the profile, homepage, matches, teams, and adding comments to the review page. The components category was also finished during these sprints, as there were a few components that were specific to some of these pages and were not needed to be developed previously. As every other task finished, the development ended with the deployment of the project to the AWS.

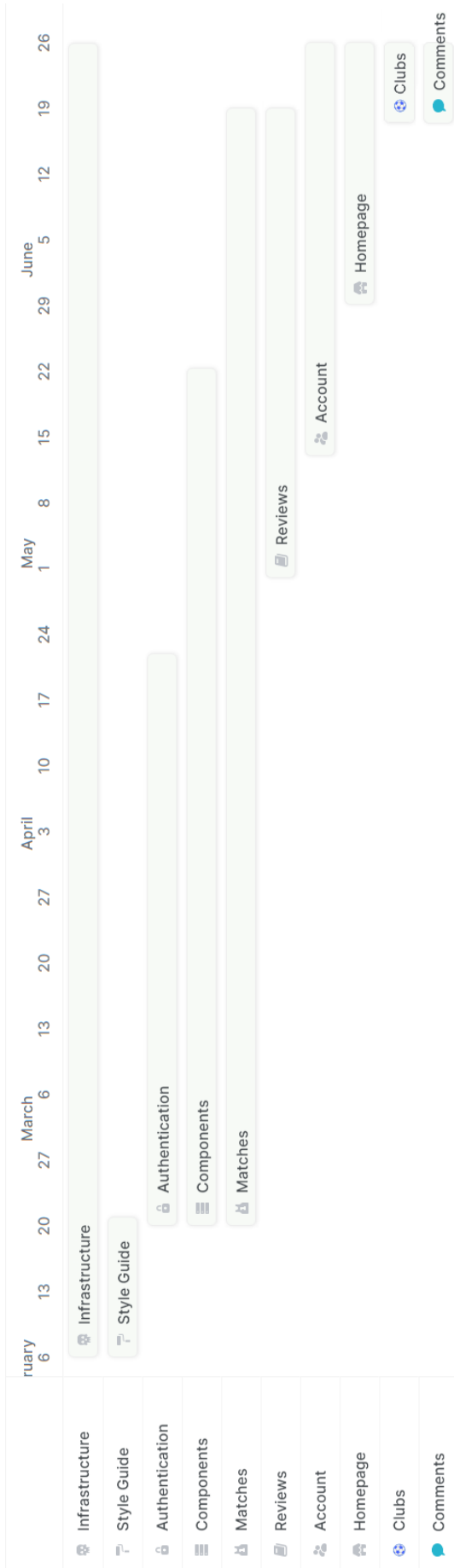


Figure 7.2: Internship development timeline

7.4 Developed Functional Requirements

This section provides an overview of the completion status of the functional requirements defined in Section 3.3. The requirements prioritized as "Won't have" are not included in the following tables, as none of them were completed. The Experiences (FR-54 to FR-74), football statistics (FR-84 and FR-85), and previous seasons (FR-86), will also not be featured, as they were removed from the project's scope, as described in Section 7.2. These exclusions help to keep the tables concise and focused.

The structure of the following tables is similar to the ones in Section 3.3, but the actor columns have been replaced with a "Completed" column. A "Y" in the "Completed" column indicates that the requirement has been completed, while an "N" signifies that it is yet to be fulfilled.

Area	Id	Requirements	Priority	Completed
Authentication	FR-1	Login	Must ▼	Y
	FR-2	Logout	Must ▼	Y
	FR-3	Register	Must ▼	Y
	FR-4	Recover password	Must ▼	Y
	FR-5	Change password	Must ▼	Y
	FR-6	Remember login information	Could ▼	Y
Profile	FR-7	See and edit your profile picture	Must ▼	Y
	FR-8	See and edit personal information	Must ▼	Y
	FR-9	Follow an author	Must ▼	Y
	FR-10	See frequent content level	Could ▼	N
	FR-11	See frequent tags	Could ▼	N
	FR-12	See joining date	Could ▼	Y
	FR-13	See total likes, comments and shares	Could ▼	Y
	FR-14	See number of followers	Could ▼	Y
Collection management	FR-15	See number of followed users	Could ▼	Y
	FR-17	Create collections	Must ▼	Y
	FR-18	Delete collections	Must ▼	Y
	FR-19	Add collection items	Must ▼	Y
	FR-20	Remove collection items	Must ▼	Y
	FR-21	See collection items	Must ▼	Y
	FR-22	Set and change collection name	Should ▼	Y
Collection list	FR-23	Set and change collection visibility	Should ▼	Y
	FR-24	See author's collections	Must ▼	Y
	FR-25	Filter collection list	Could ▼	N

Table 7.1: Completed requirements - Authentication and Profile

Area	Id	Requirements	Priority	Completed
List Reviews	FR-26	List all reviews	Must ▼	Y
	FR-27	Search review list by text	Should ▼	Y
	FR-28	Sort review list	Could ▼	N
	FR-29	Apply filters to review list	Should ▼	Y
	FR-30	Show trending reviews	Could ▼	Y
Review	FR-31	Read full review	Must ▼	Y
	FR-32	See associated match	Must ▼	Y
	FR-33	See review statistics	Could ▼	N
	FR-34	Filter review statistics	Could ▼	N
	FR-35	See a specific review's details	Should ▼	Y
	FR-36	Share on social media	Could ▼	N
	FR-37	Comment review	Should ▼	Y
	FR-38	Like the review	Must ▼	Y
Tags	FR-40	List related tags	Could ▼	N
Add review	FR-41	Choose match to review	Must ▼	Y
	FR-42	Change associated match	Should ▼	Y
	FR-43	Add a title	Must ▼	Y
	FR-44	Add a body	Must ▼	Y
	FR-45	Add, delete and replace images	Must ▼	Y
	FR-46	See image gallery	Could ▼	N
	FR-47	Add thumbnail	Should ▼	Y
	FR-48	Add language	Should ▼	Y
	FR-49	Add content level	Should ▼	Y
	FR-50	Add tags	Must ▼	Y
	FR-51	Preview reievw	Should ▼	Y
	FR-52	Publish review	Must ▼	Y
	FR-53	Save draft	Should ▼	Y
	List matches	FR-75	See matches	Must ▼
FR-76		Search matches by text	Should ▼	Y
FR-77		Apply filters to match list	Should ▼	Y
Match	FR-78	See match information	Must ▼	Y
	FR-79	See match details	Could ▼	Y
	FR-80	Mark match as watched	Should ▼	Y
	FR-81	Like match	Must ▼	Y
Team	FR-82	See team information	Must ▼	Y
	FR-83	Follow team	Must ▼	Y

Table 7.2: Completed requirements - Content and Football

The tables demonstrate that all requirements categorized as "Must have" and "Should have" have been completed. This was due to the project readjustment and reprioritization, which took place during the development phase. As a result, the project's scope was reduced, and the priority of the Experiences requirements was downgraded to "Won't have".

This evidence shows that the adjustments made during development were effective. Despite the time constraint imposed by the internship, only a few "Could have" requirements were marked as not completed. These unfulfilled requirements were primarily statistics, filters and a few non-essential pages, all of which were assigned the second lowest priority.

In conclusion, the completion of the crucial requirements, in conjunction with the insignificant impact derived from the unfinished ones, lead to a positive outcome for the project.

7.5 Developed Architecture

The changes to the scope of the internship, described in Section 7.2, along with the developed requirements, listed in Section 7.4, have resulted in alterations to the project's final architecture and entity-relationship diagrams.

7.5.1 Developed C4 Model

This subsection will present the developed version of the C4 Models, alongside the differences between the developed versions and the planned versions, created in Section 4.2.

To improve the clarity and understanding of the changes in each C4 diagram, the removed elements are highlighted in red. This will help to easily identify the elements that have been removed from the original diagrams.

Developed Context Diagram

This diagram remains the same except for the API Football, which was removed as it was not used.

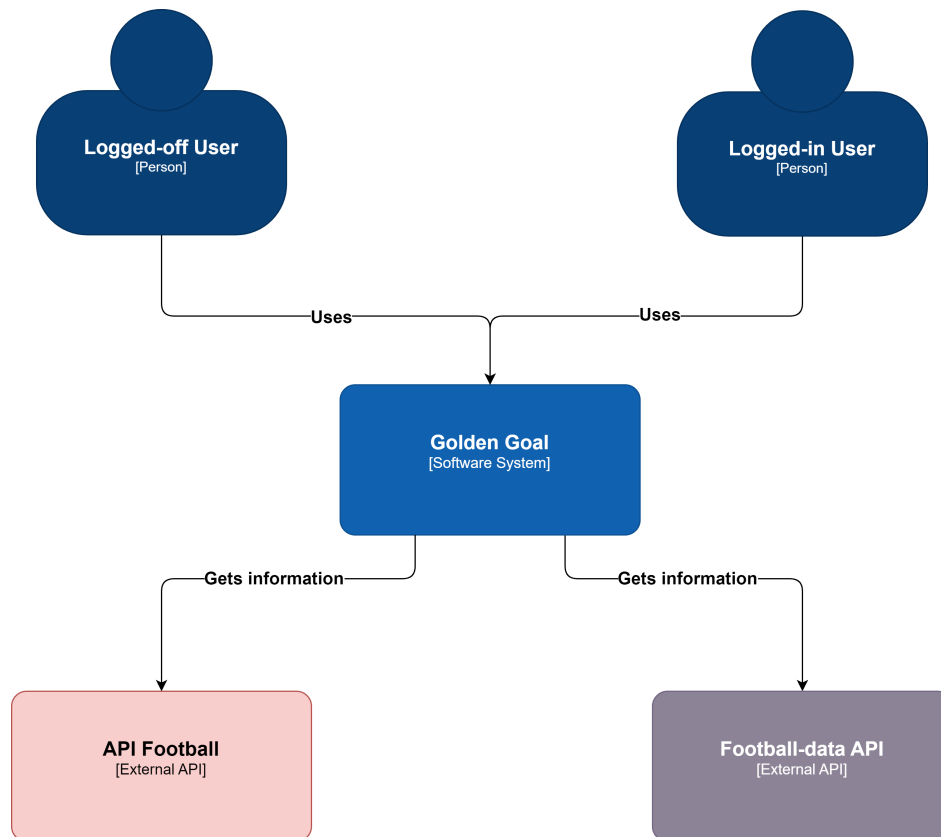


Figure 7.3: Developed context diagram

Developed Container Diagram

There were two alterations to the platform’s container diagram. The internal load balancer and the API Football were removed, as the load balancer was not created and the API was not used.

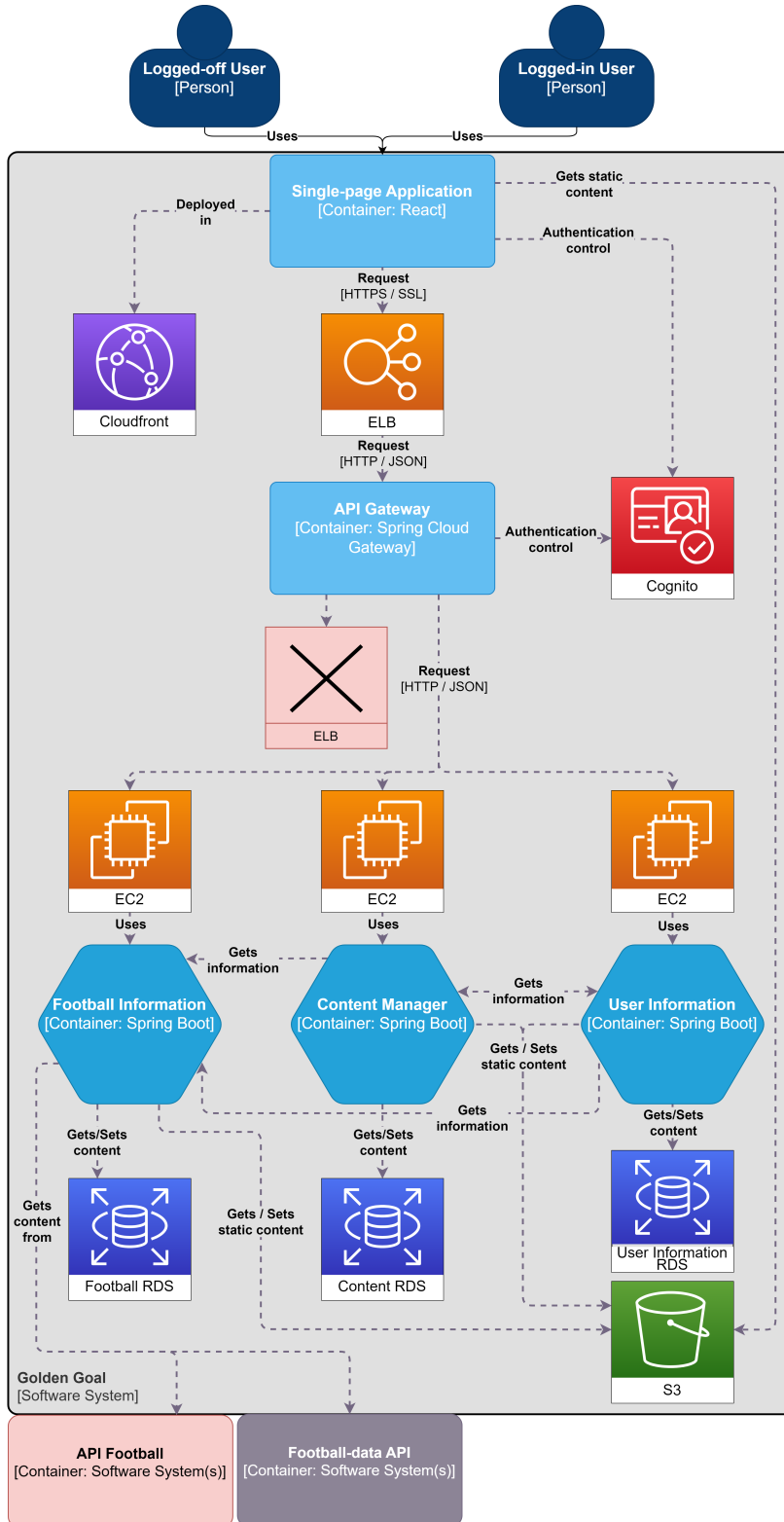


Figure 7.4: Developed container diagram

Developed Single-Page Application Component Diagram

In this diagram only the Experiences Views and Experiences State were removed, as they were removed from the internship’s development scope.

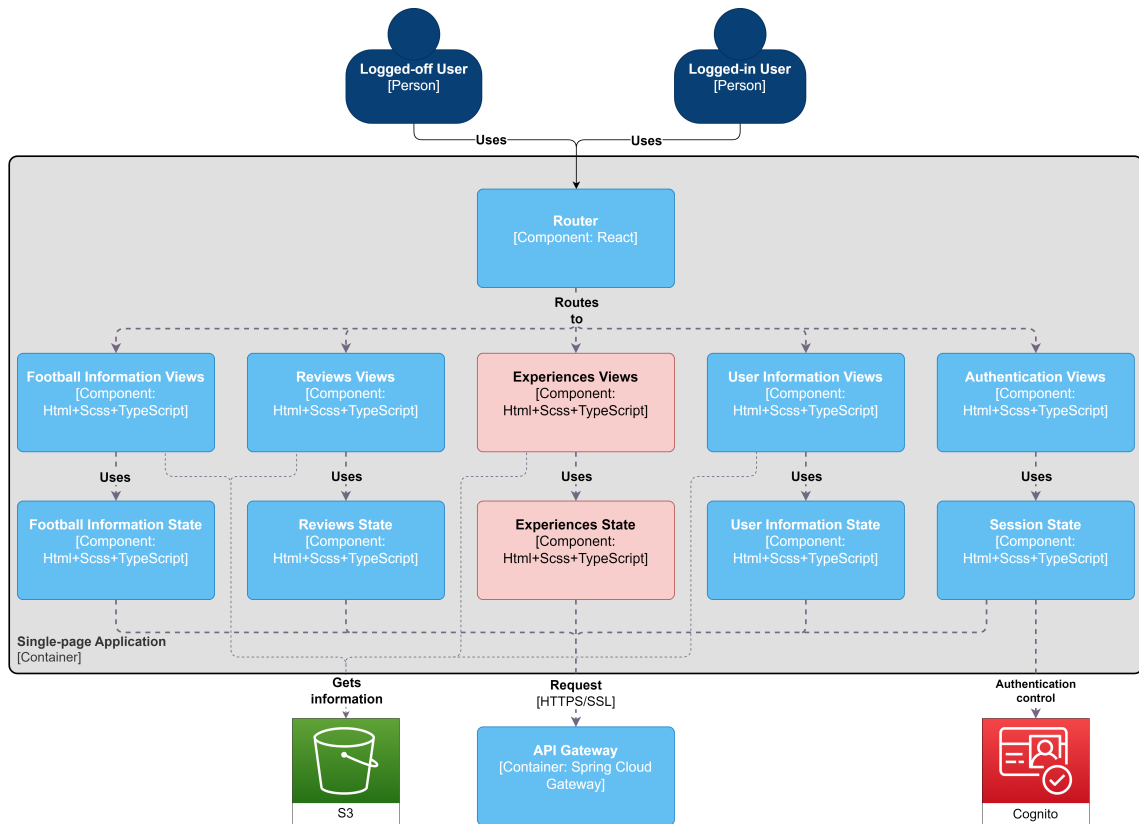


Figure 7.5: Developed Single-Page Application component diagram

Developed Football Information Component Diagram

This diagram was changed by removing the Statistics and People components, both of which were not created. As these were the only two components using API Football, this API and its connector were also removed.

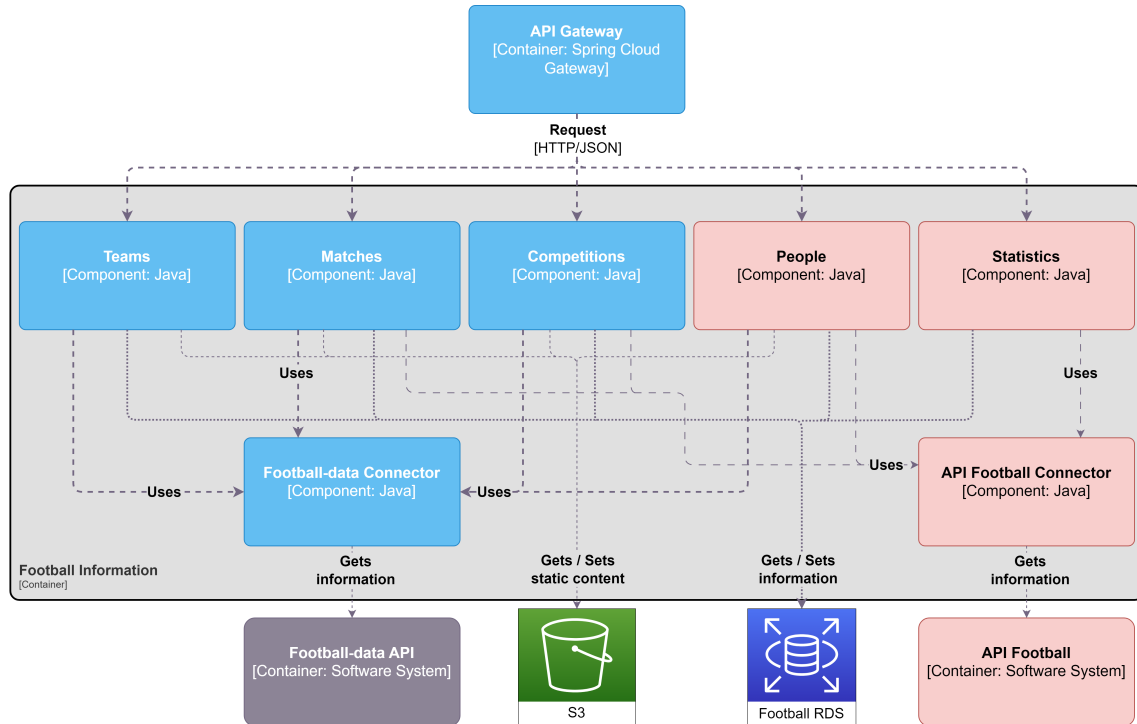


Figure 7.6: Developed Football Information component diagram

Developed Content Manager Component Diagram

The Content Manager component diagram was one of the most altered, as neither the Experiences component nor the Statistics component were developed during the internship.

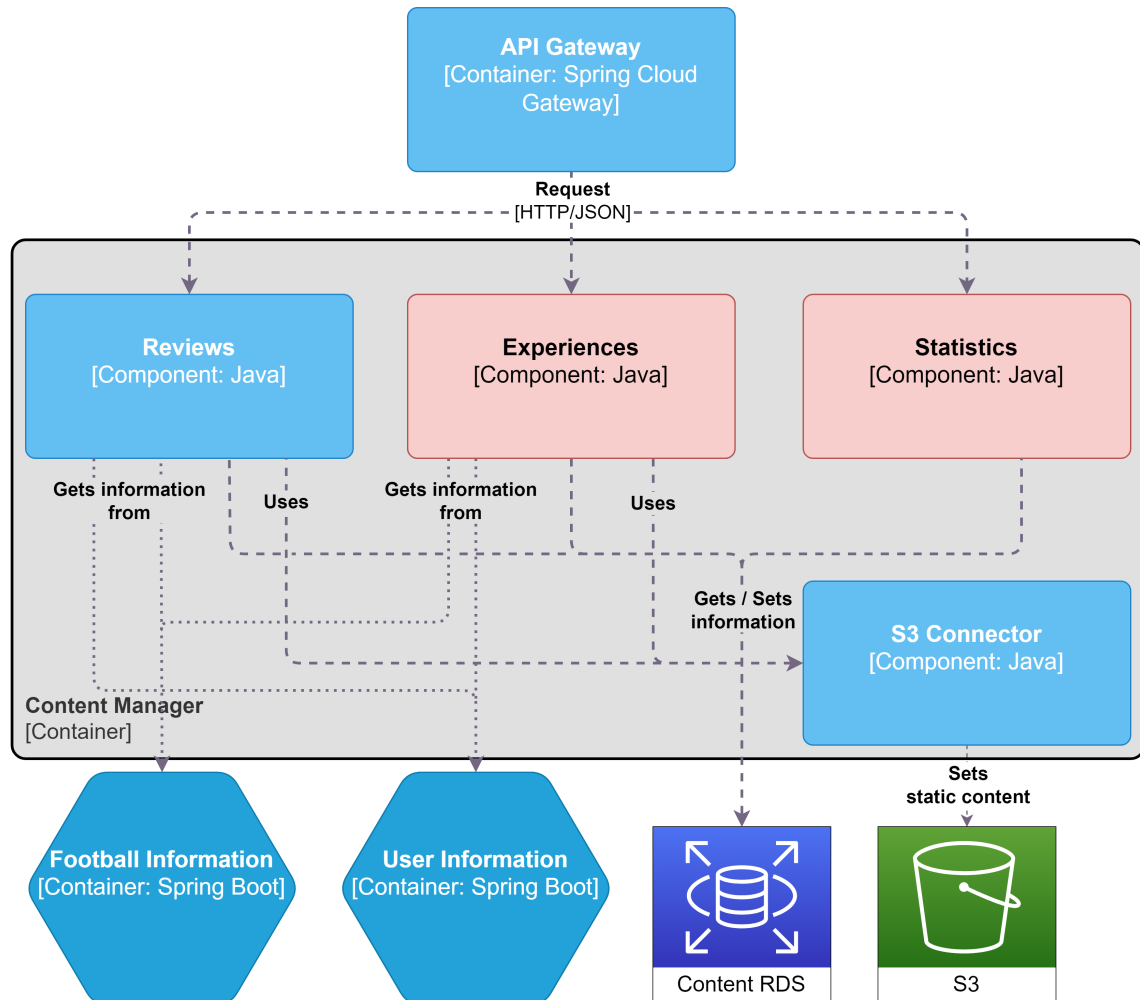


Figure 7.7: Developed Content Manager component diagram

Developed User Information Component Diagram

The User Information component diagram did not suffer any alterations.

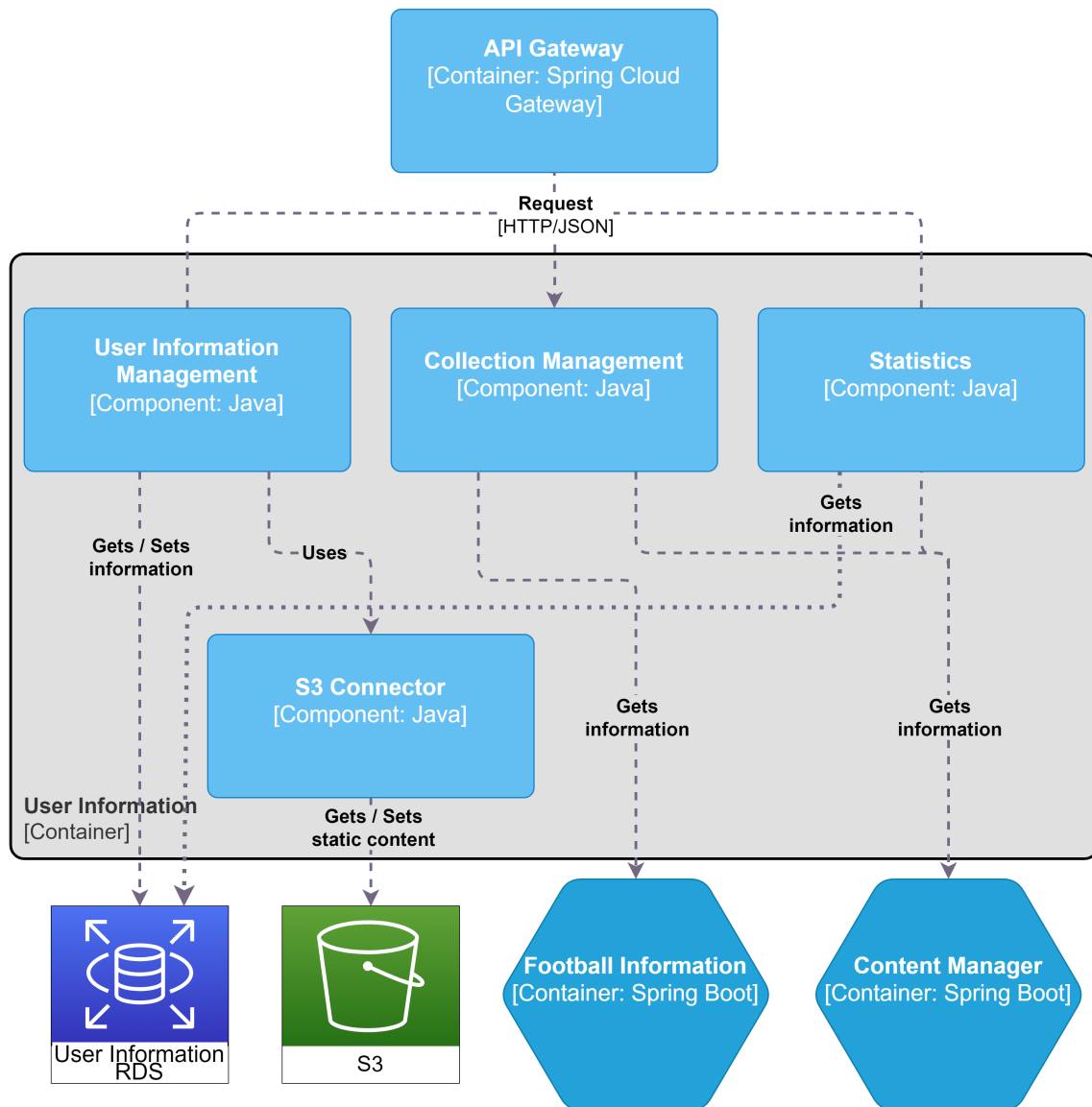


Figure 7.8: Developed User Information component diagram

Developed API-Gateway Component Diagram

Similarly to the previous diagram, this one remained unaltered.

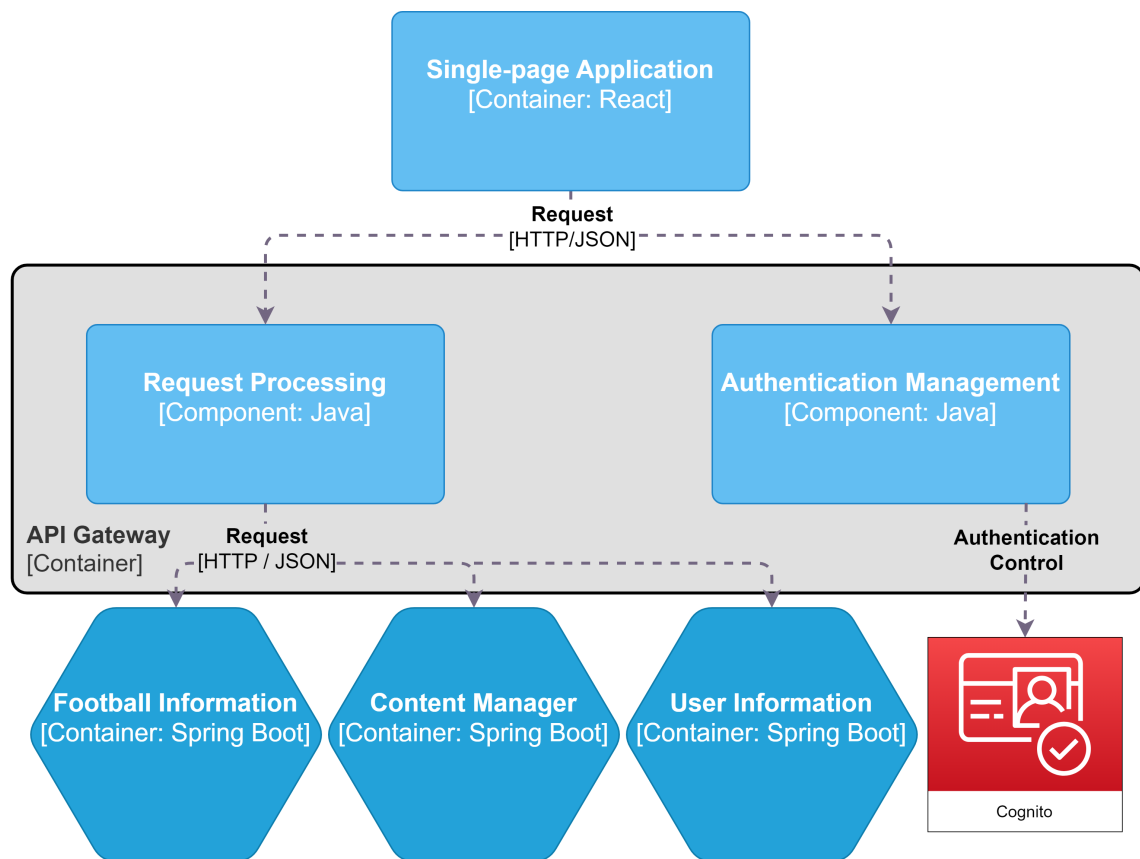


Figure 7.9: Developed API-Gateway component diagram

7.5.2 Developed Entity-Relationship Diagram

This subsection list the alterations made to the planned versions of the entity-relationship diagrams created in Section 4.3, along with the final versions of each diagram.

Developed Football Information Diagram

This diagram underwent two significant changes, when compared to the original version. The first was the removal of the "editions" table, and the second was the removal of "people" and related tables, such as "managers", "players", and "people_careers".

On the other hand, the "matches", "teams", and "competitions" tables were expanded, in order to accommodate data that was not considered during the planning.

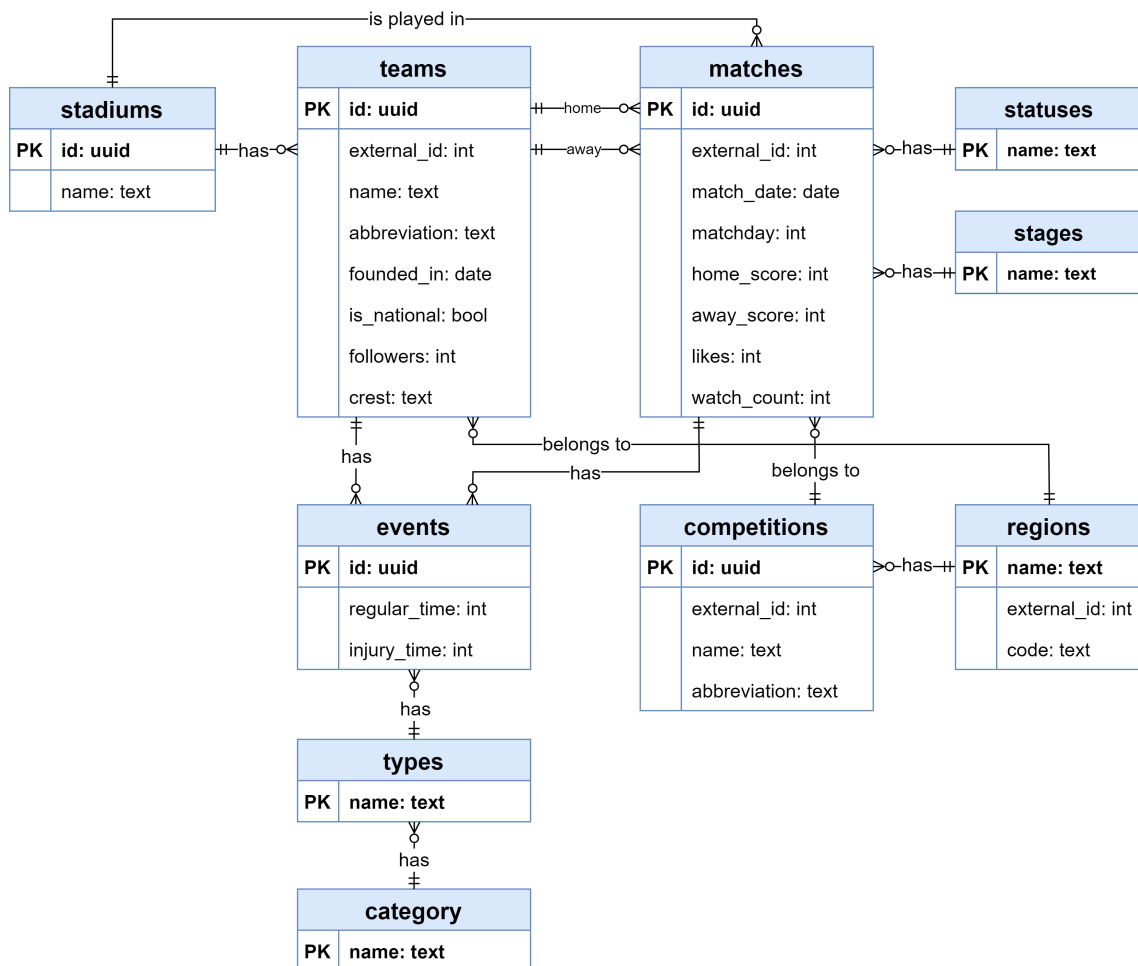


Figure 7.10: Football Information developed entity-relationship diagram

Developed Content Manager Diagram

The Content Manager diagram had the "experiences" and related tables removed, as they were excluded from the project's scope. The "reviews" table has been expanded to store reviews' titles, thumbnails, likes, and drafts.

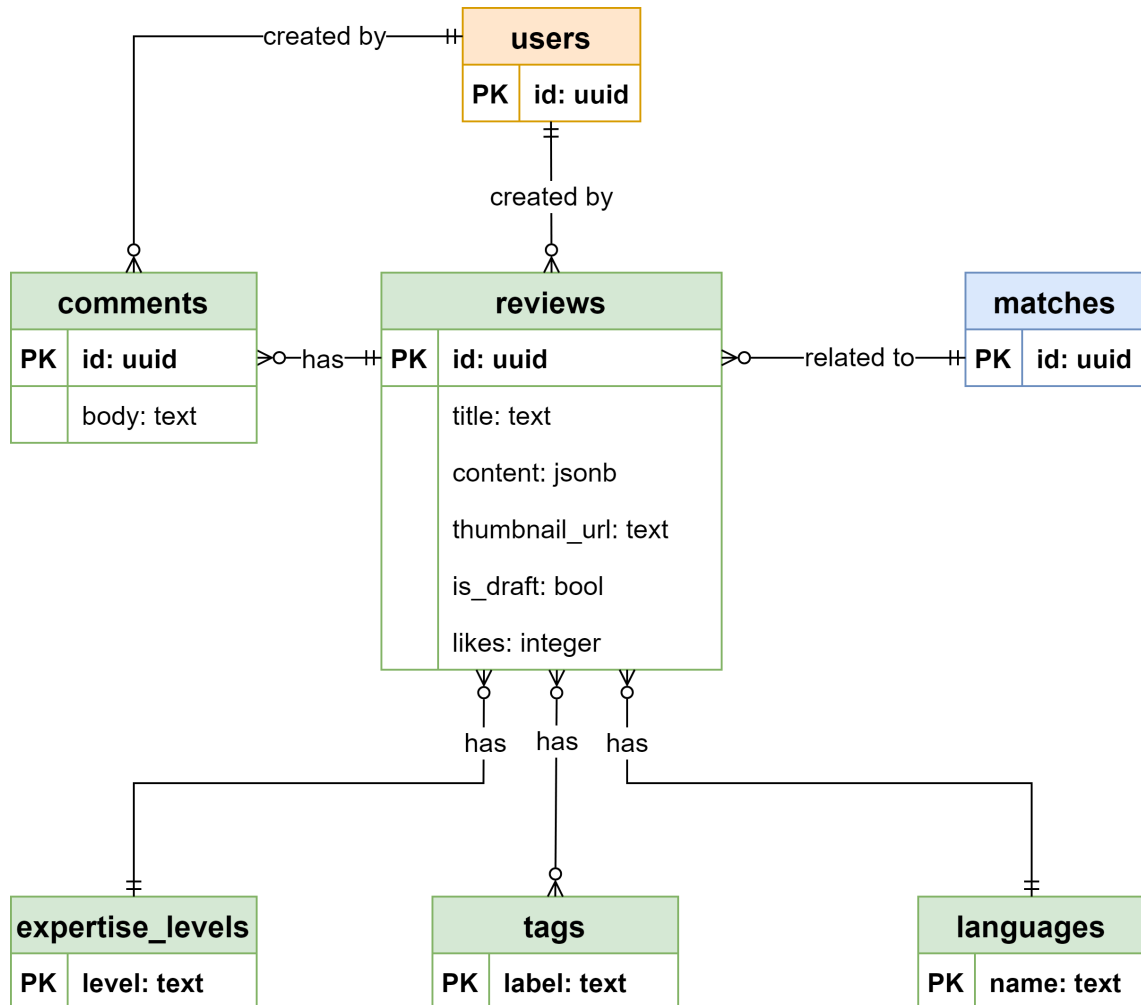


Figure 7.11: Content Manager developed entity-relationship diagram

Developed User Information Diagram

The final version of this diagram is very similar to the original one. The "experiences" relationships were removed, due to these tables not existing. The follower count was added to each member of the "users" table, alongside the profile picture url, and a new table, "content_types", was created, to distinguish between match collections and review collections.

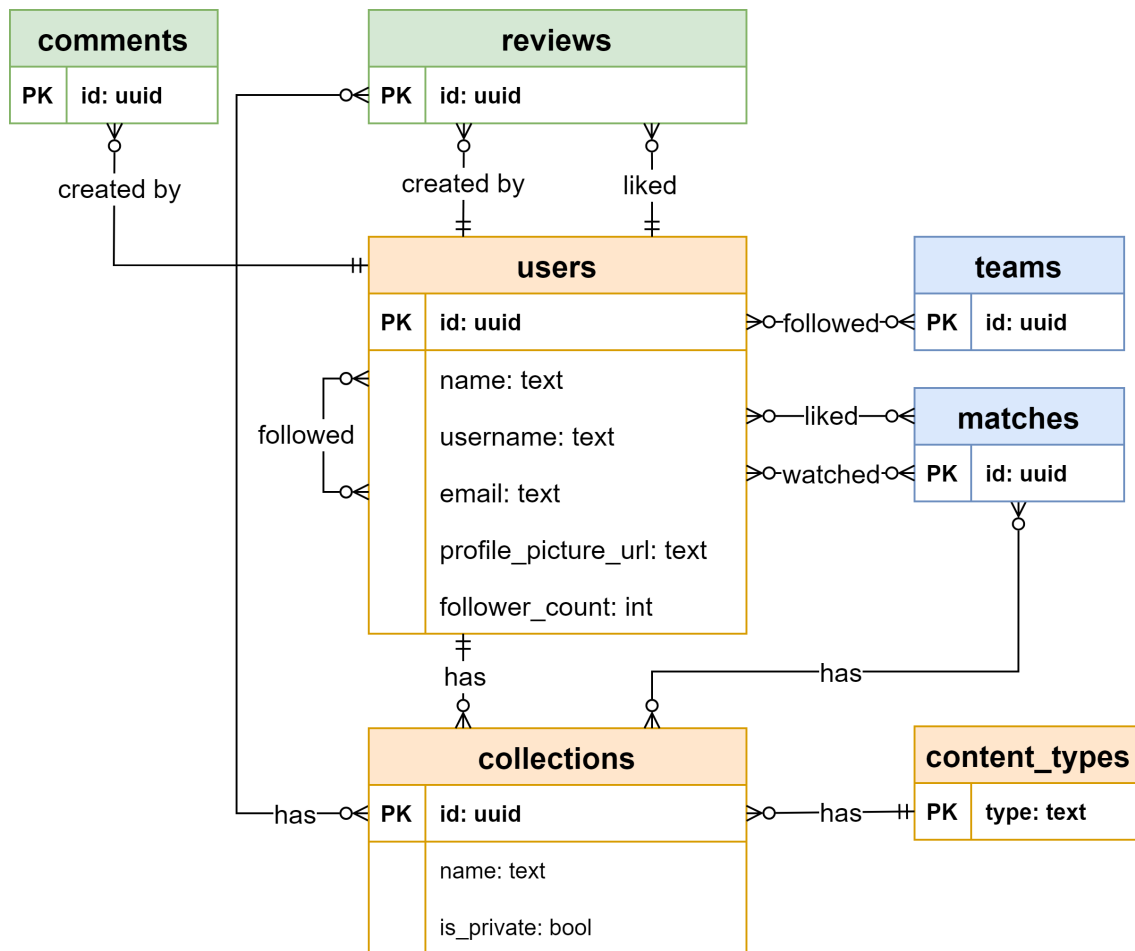


Figure 7.12: User Information developed entity-relationship diagram

7.6 Platform Functionalities

This section serves as an exhibition of the developed pages. Due to the extensive number of screens involved, only the main screens will be presented in this section. A complete list of all screens can be found in Appendix C.

7.6.1 Authentication

The following two images belong to the Authentication module of the platform. Figure 7.13 shows the login component of the authentication page, while Figure 7.14 showcases the registration component.

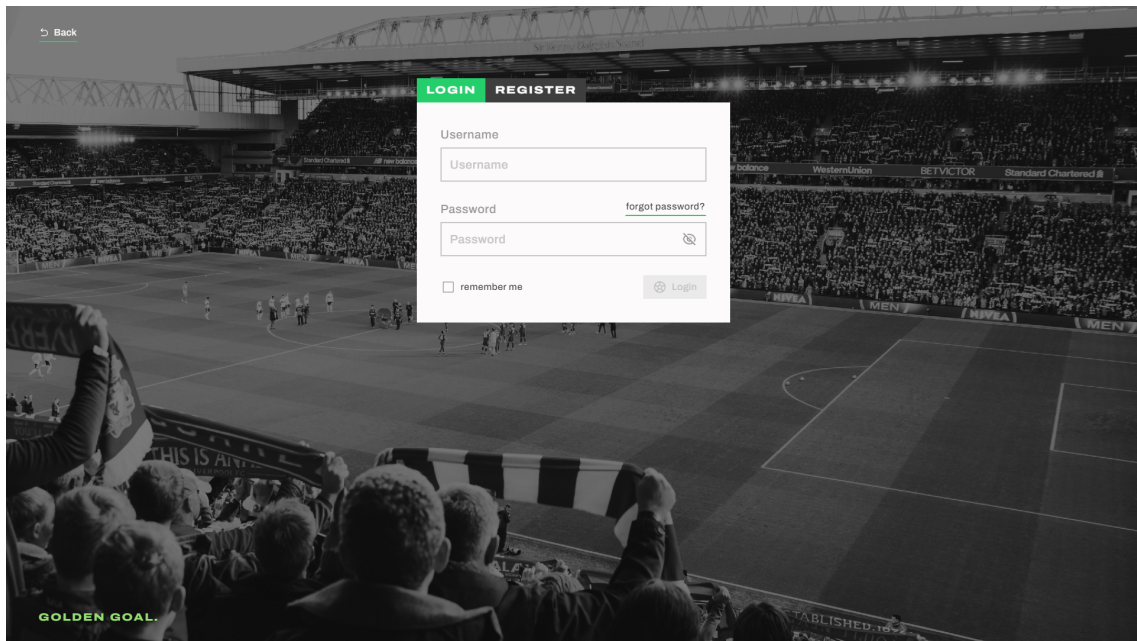


Figure 7.13: Login page

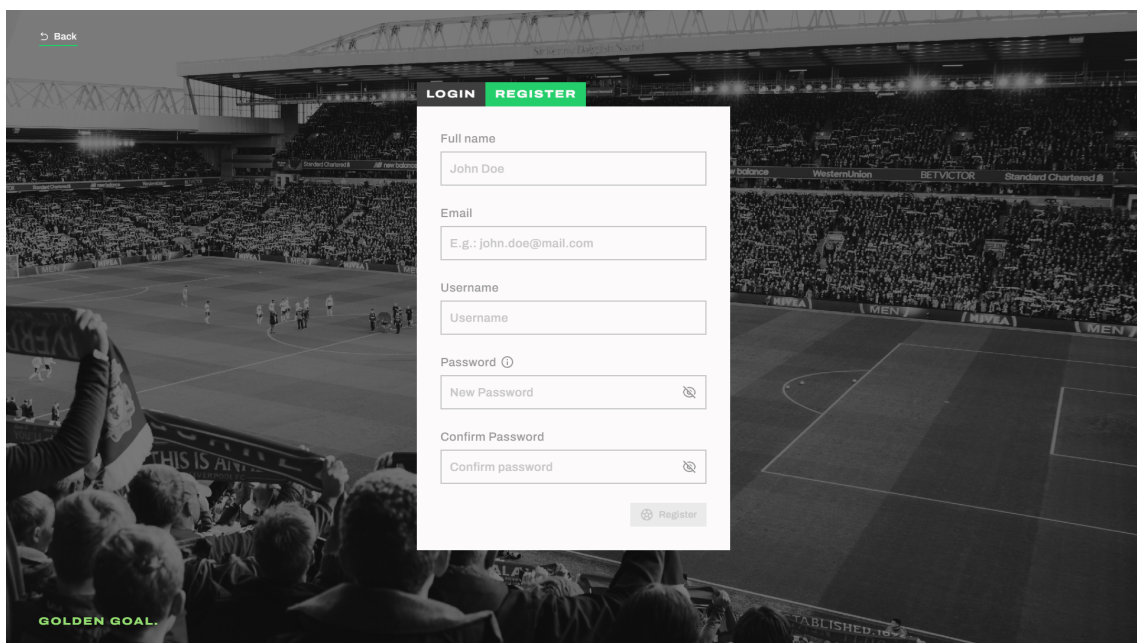


Figure 7.14: Registration page

7.6.2 Profile

The next three figures show the private profile of a user, which has three different tabs. The My Data tab, represented in Figure 7.15, allows the user to see and edit his own data. The Reviews tab, seen in Figure 7.16, lists the user's reviews and drafts, both of which are editable. Lastly, the Collections tab displayed in Figure 7.17 lists the user's collections, where the user can add a collection and manage existing ones.

The public profiles, presented in Appendix C, have a few differences from the private ones, mainly the absence of the "My Data" tab, the non-editability of the reviews and collections, and the follow button.

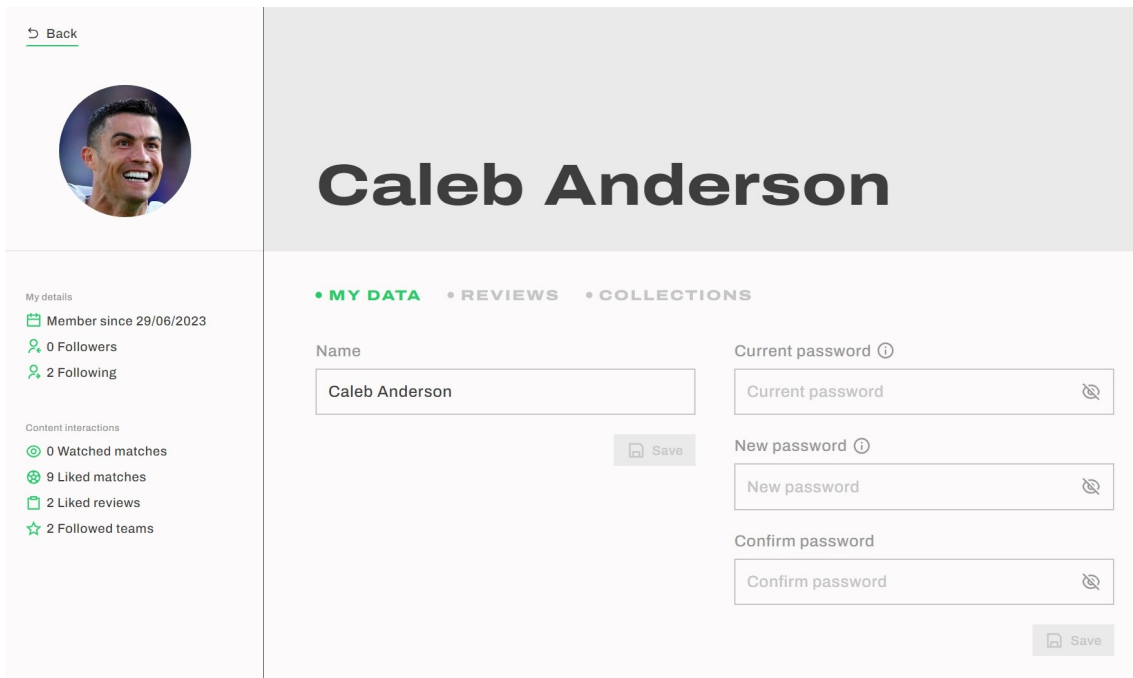


Figure 7.15: Profile - My Data tab

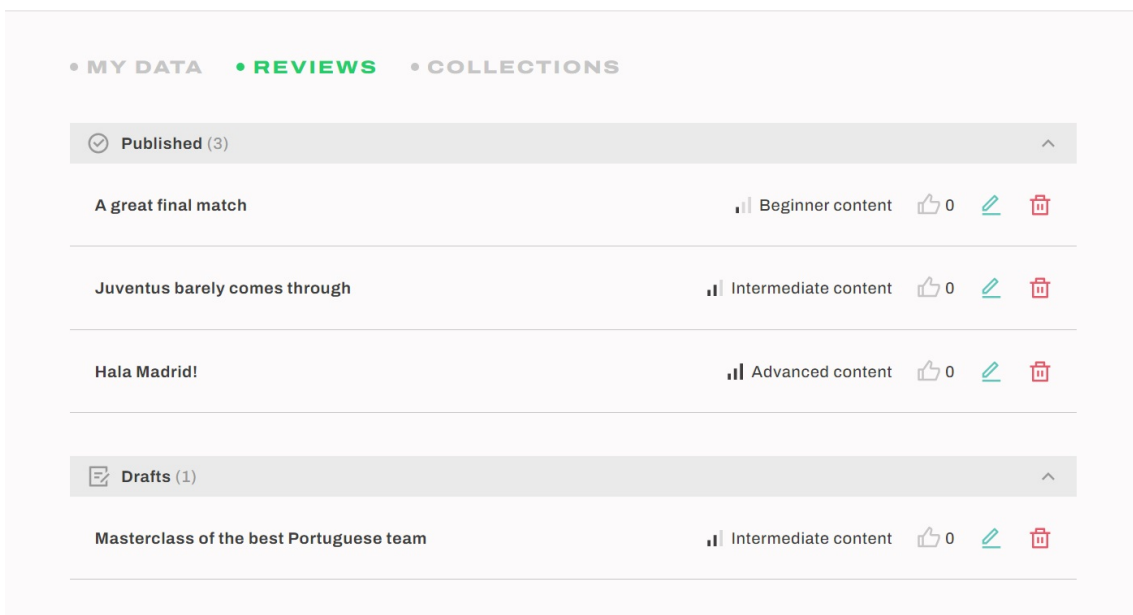


Figure 7.16: Profile - Reviews tab

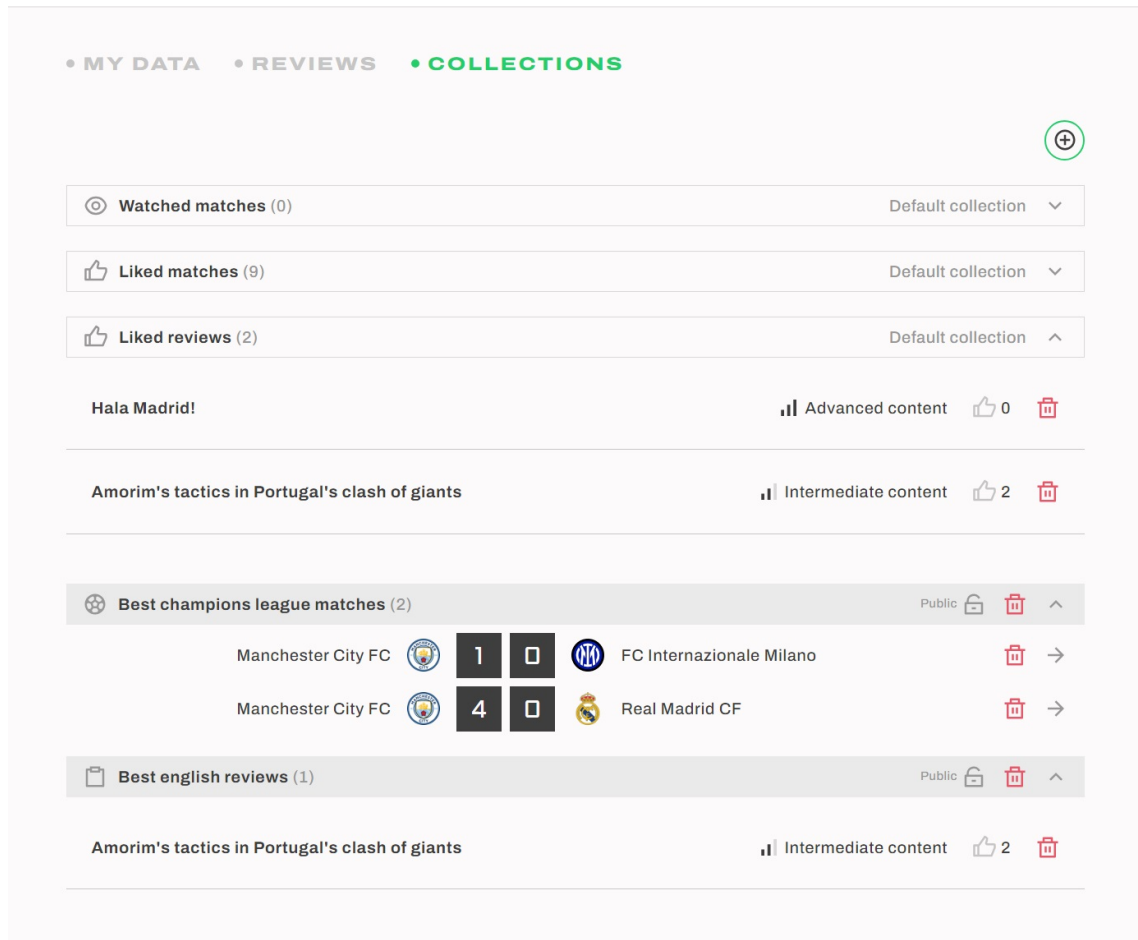


Figure 7.17: Profile - Collections tab

7.6.3 Reviews

This subsection is composed by two different pages. The first one, as seen in Figure 7.18, allows the user to view and filter all the reviews available in the platform, alongside showing popular reviews and authors.

Figure 7.19 showcases a review page. This page includes the review's title, tags, content, match information, and author details. This page also includes interaction buttons on the top right corner, where a user can add it to a collection or like it. It is followed by a comment section and by suggested reviews from the same match, as seen in Figure 7.20. Even though both these figures belong to the same page, it has been divided into two images to improve visibility.

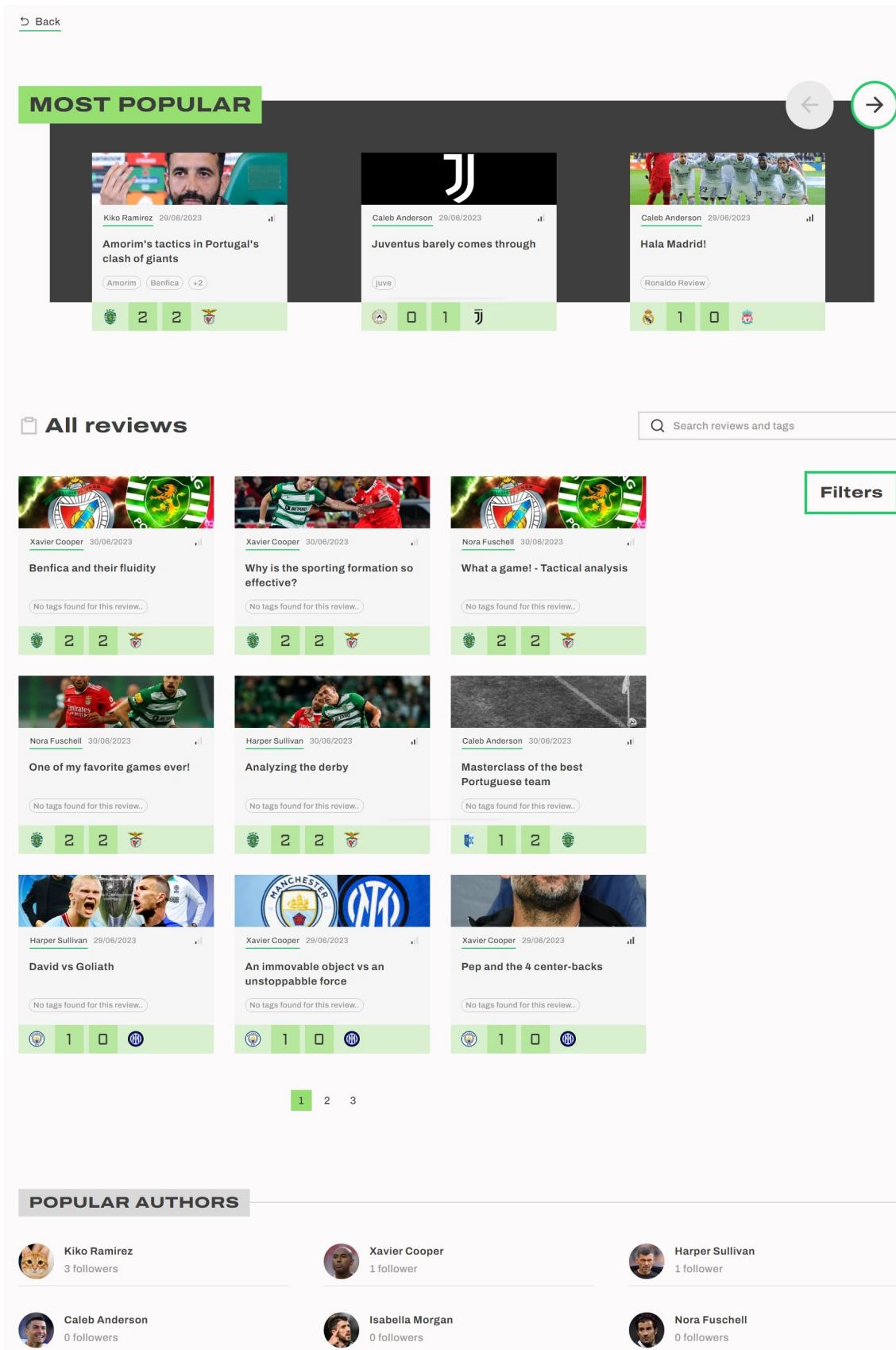



Figure 7.18: Review list page


[Back](#)



Primeira Liga 21/05/2023
Estádio José Alvalade


Review details

Intermediate content 2




Kiko Ramirez
Member since 29/06/2023
3 followers

[Unfollow](#)



Rúben Amorim's tactical approach against Benfica showcased his ability to adapt and strategize effectively against strong opponents. Sporting CP, under Amorim's guidance, displayed a compact and disciplined defensive structure that proved difficult for Benfica to break down. The team defended as a cohesive unit, with players maintaining their positions and applying intense pressure on the ball carrier. This defensive solidity limited Benfica's attacking opportunities and frustrated their attempts to create meaningful chances.




EBL · Mar 8, 2023
@EBL2017 · [Follow](#)

Rúben Amorim of Sporting is regarded by many as the next big thing in coaching. I can understand why to an extent as he is clearly a talented coach.

However, Arteta's Arsenal are another level. Sporting could win tomorrow, but over the two legs, that's HIGHLY unlikely.

THREAD!



EBL · Mar 8, 2023
@EBL2017 · [Follow](#)

I was not familiar with Rúben Amorim before this thread, but I must admit that I am underwhelmed with his tactical qualities relative to the reputation he has...

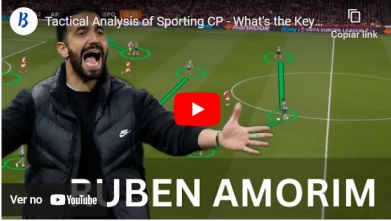
Sporting play a 3-4-3 system similar to that of Conte at Spurs except with a supposedly more 'proactive' philosophy.

4:24 PM · Mar 8, 2023

328 · [Reply](#) · [Share](#)

[Read 2 replies](#)

Amorim's tactical flexibility was also evident in his in-game adjustments. As the match progressed, he made shrewd substitutions and alterations to the team's approach, ensuring that Sporting CP maintained control and minimized Benfica's impact. This adaptability showcased Amorim's tactical astuteness and his ability to read the game effectively.



Overall, Rúben Amorim's tactics against Benfica highlighted his strategic prowess and ability to create a well-organized and disciplined team. Sporting CP's defensive solidity, effective counter-attacking, and tactical flexibility were instrumental in their success against a strong opponent. Amorim's innovative methods and adaptability continue to make him one of the most promising young coaches in football today.

Written by [Kiko Ramirez](#) on 29/06/2023

Figure 7.19: Review page - review content

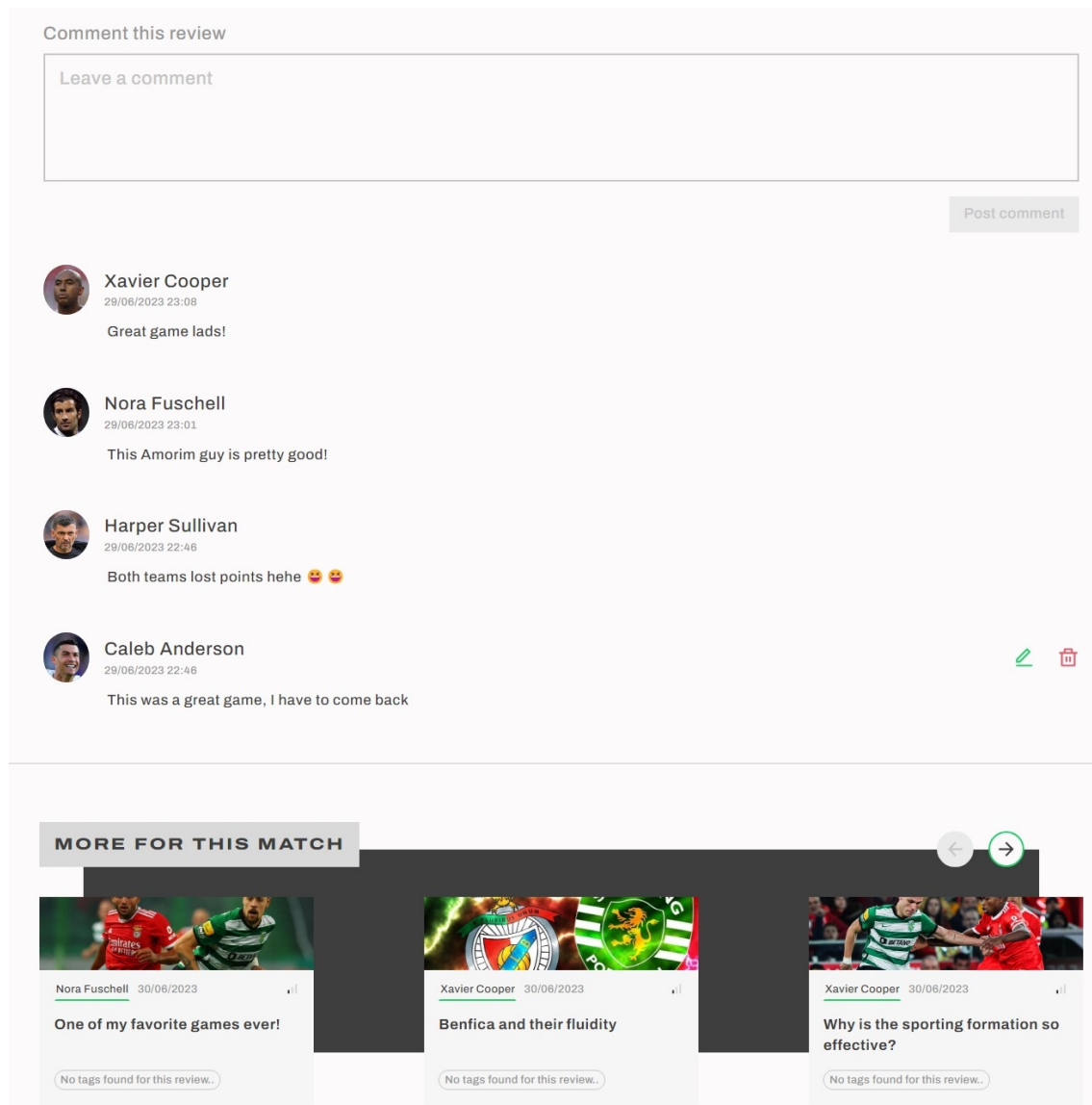


Figure 7.20: Review page - comments and suggested reviews

7.6.4 Review Creation

Figure 7.21 shows the page responsible for writing a review. Using a rich text editor, the user can freely format and insert content into the review. As can be seen in Figure 7.22, this text editor allows the user to:

- Undo / redo changes;
- Change text type and style (headings, bold, ...);
- Change text color;
- Create bullet points, numbered lists, checklists, and quotes;
- Insert links, tables, images (with captions), Tweets, and YouTube videos;

- Align and indent text;
- Write using speech-to-text.

This page also allows to set the review title and details, such as language, expertise level, tags, and thumbnail.

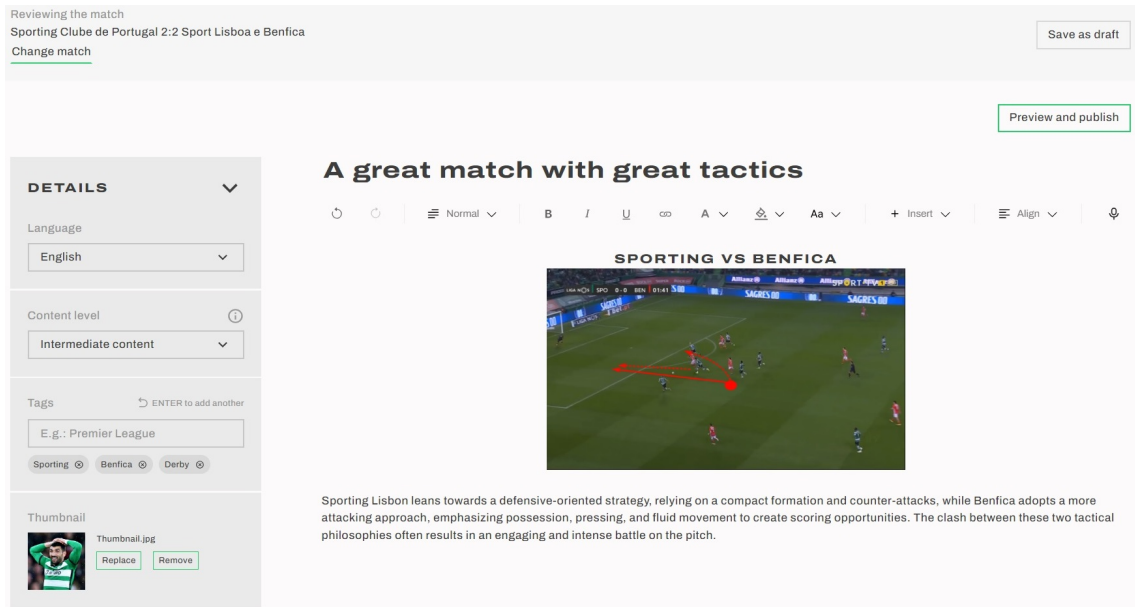


Figure 7.21: Create review page

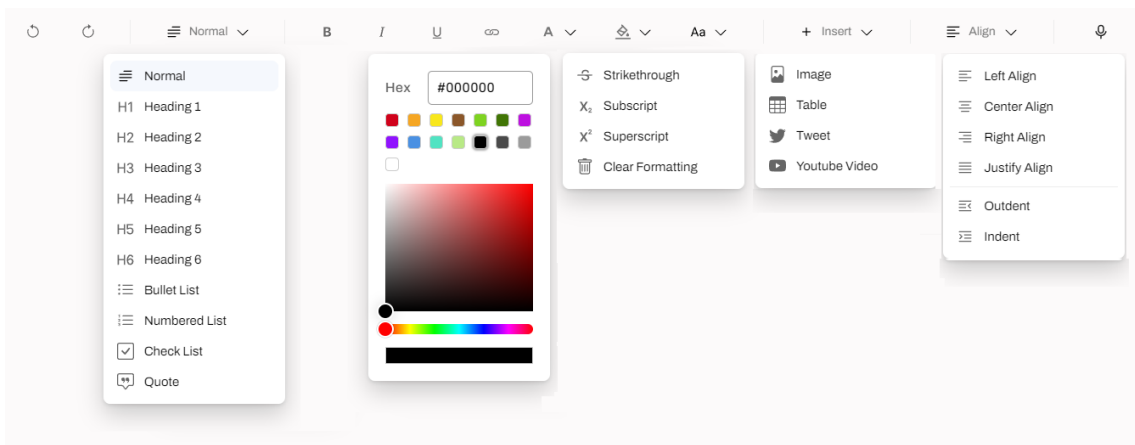


Figure 7.22: Create review toolbar expanded

7.6.5 Matches

The Matches module contains two main pages. The Match List page, showcased in Figure 7.23, shows a list of all matches, ordered by date and competition. This list can be filtered by competition, league name and team name.

Figure 7.24 shows the page of an individual match, which displays the match teams, score, and details. Alongside it are displayed the most popular reviews

associated with the match and all the reviews associated with it, the last of which can be filtered by language. This page, similar to the review page, features interaction buttons, where the user can mark the match as watched, like it, or add it to a collection.

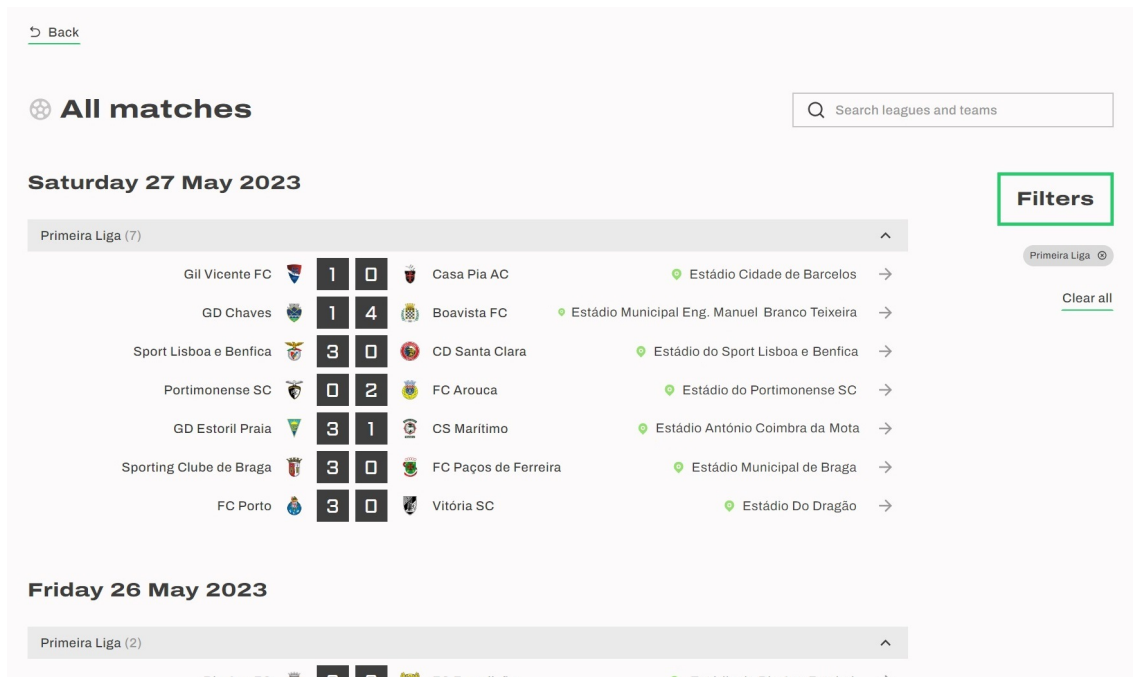


Figure 7.23: Match list page

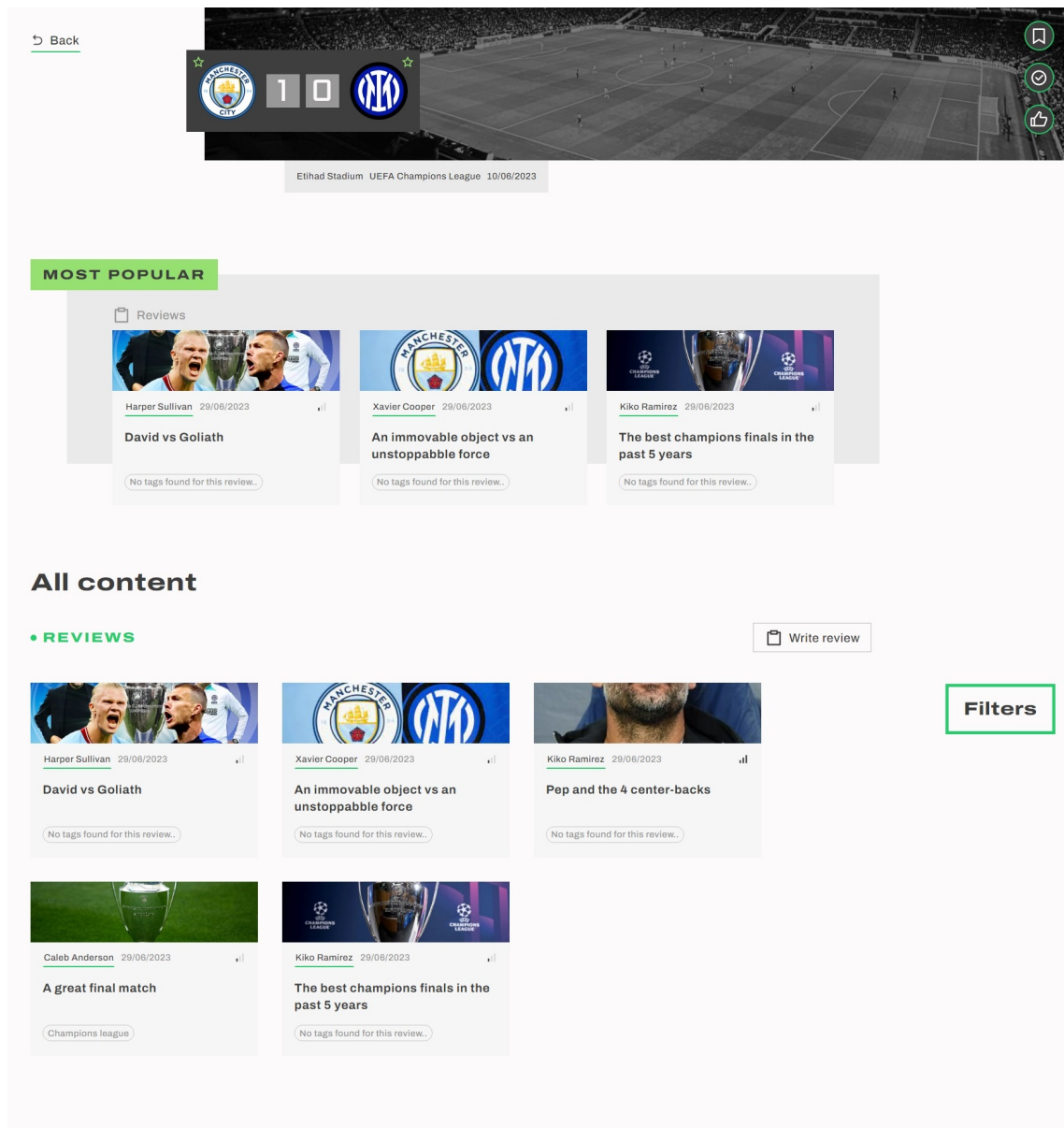


Figure 7.24: Match page

7.6.6 Team

This subsection presents the team page, which features the team name and crest, alongside information about the team and its followers.

There are two tabs present in this page, the first one being the matches tab, as seen in Figure 7.25, which lists all matches from the team grouped by competition. The second tab, displayed in Figure 7.26, lists the reviews written about the team.

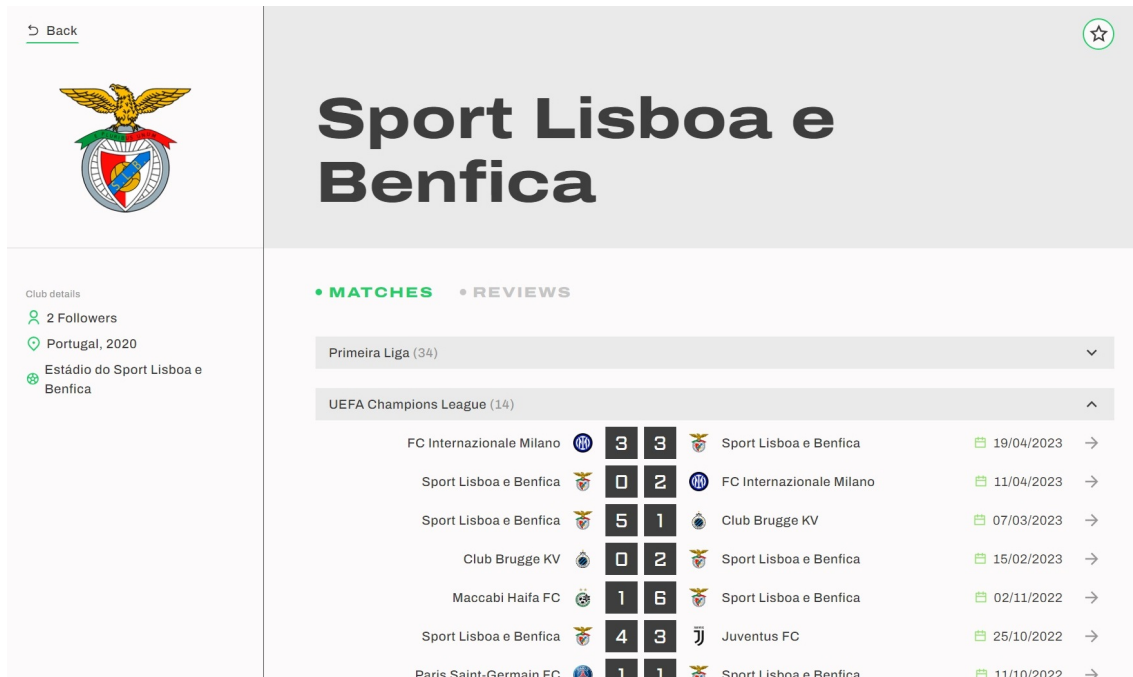


Figure 7.25: Team page - Matches tab

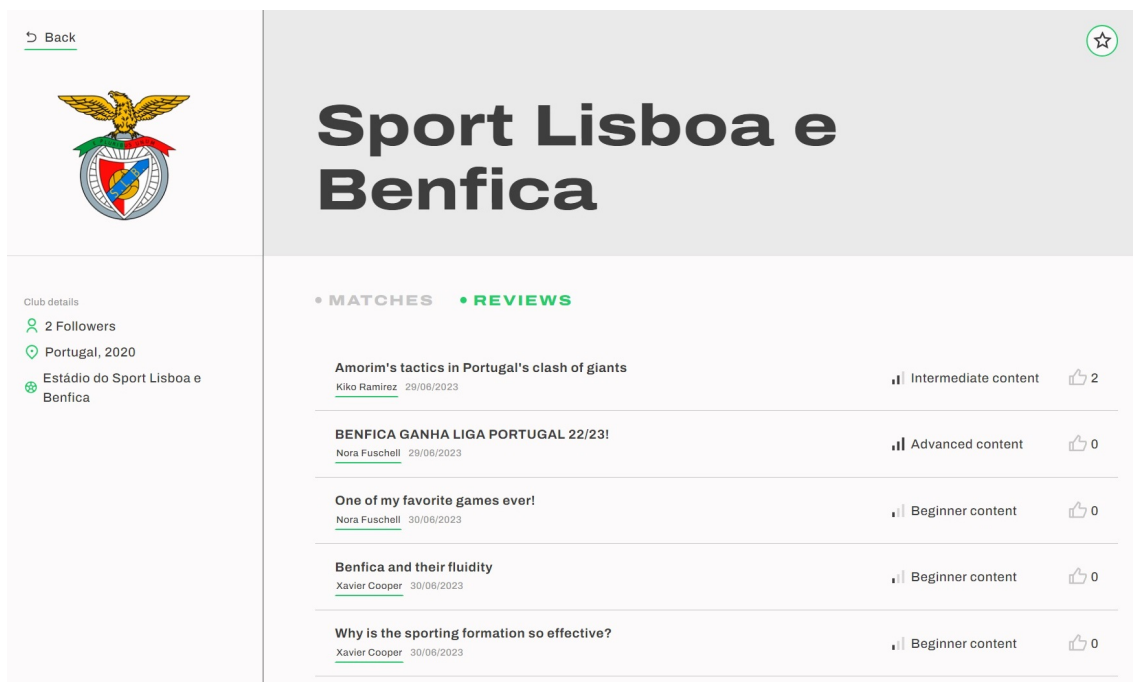


Figure 7.26: Team page - Reviews tab

7.6.7 Homepage

To finalize, this subsection presents the website's homepage. The homepage's reviews section contains the most popular reviews, alongside reviews from followed authors and the most recent reviews. In a similar fashion, the matches section contains the currently most popular match, matches from followed teams,

and the most recent matches.

Only the authenticated user's homepage will be presented in this subsection, as it is a more complex version of the unauthenticated user's homepage, which is present in Appendix C.

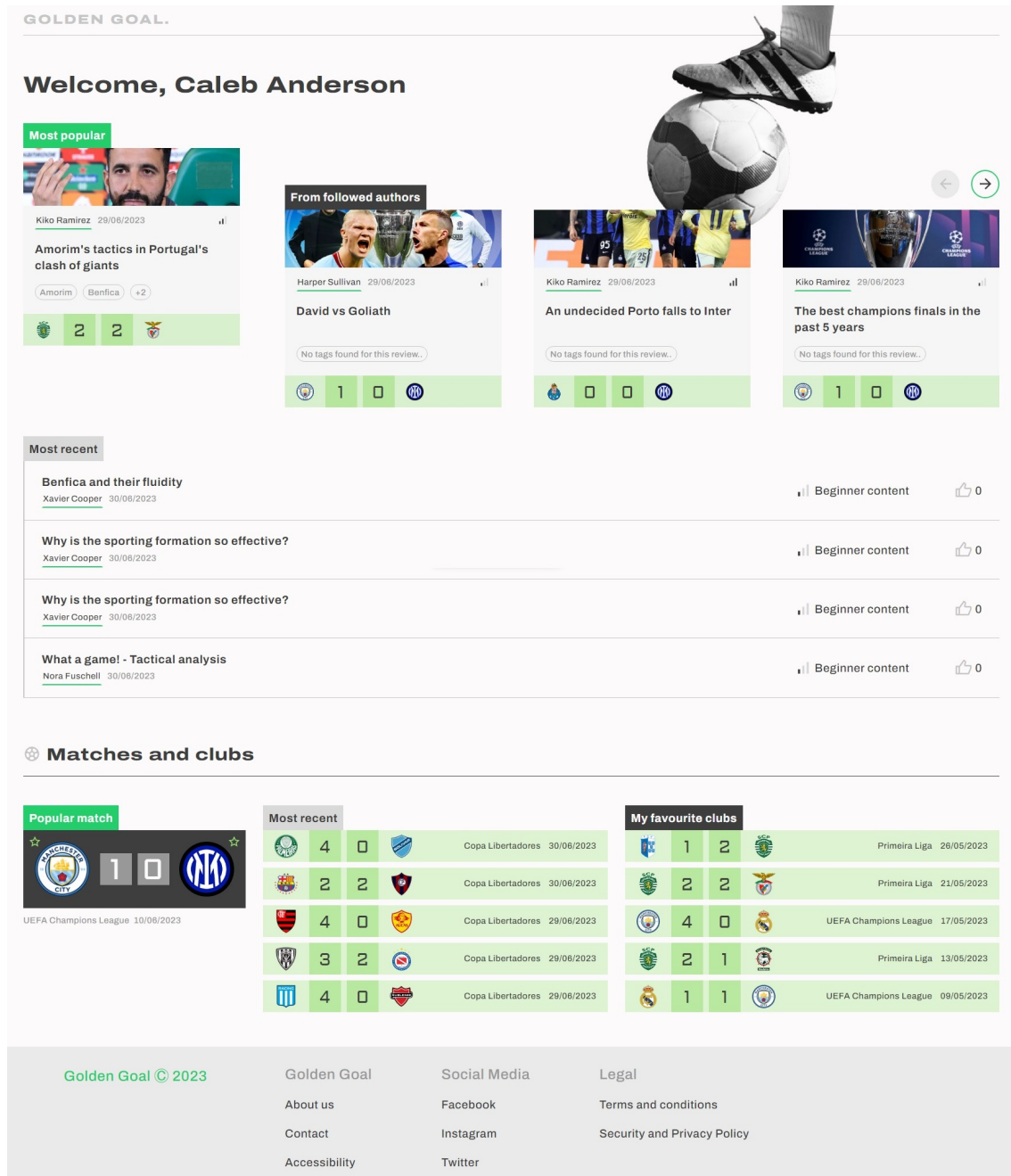


Figure 7.27: Authenticated homepage

Chapter 8

Testing

This chapter is dedicated to the testing conducted to ensure the platform's correct functioning and development.

Section 8.1 addresses the unit tests performed on the backend endpoints. Subsequently, an overview is given in Section 8.2 on the tests that were conducted to ensure the correct integration of the frontend and backend. Both of these tests try to find problems on the functional requirements, which were defined in Section 3.3.

Lastly, Section 8.3 describes the tests conducted on the non-functional requirements (NFRs), which were outlined in Section 3.4.

8.1 Unit Tests

The development of new backend endpoints was accompanied by the creation of unit tests. These tests were designed to identify errors on the newly developed endpoints, as well as any errors that might affect the existing endpoints. These tests were executed each time a pull request was created to merge code into the main branch.

The tests, created with the JUnit [119] testing framework, consisted of scripts that generated the necessary mock data for the test and then initiated a request to the backend. The request's response was then compared to an expected response, ensuring that the endpoint was functioning correctly for that specific case.

The unit tests were developed with the intention of examining as many code paths as possible, testing both success and error cases of endpoints. This approach aligns with Control Flow Testing [120], a white-box technique that aims to explore the potential code execution paths, testing as many of them as possible. Therefore, multiple tests were created for the same endpoint, to verify all feasible paths.

Figure 8.1 shows an example of a unit test, which aims to test the "Get Team by Id" endpoint.

```
@Test
void testGetTeamById() {
    log.info("Testing endpoint - Get team by id - /teams/{id}");

    // Create team and region (Region is mandatory on Team)
    competitionsDatabaseService.saveRegion(REGION MOCK);
    Team team = teamsDatabaseService.saveTeam(TEAM MOCK);

    // Send the request to the backend
    ParameterizedTypeReference<TeamResponse> responseType
        = new ParameterizedTypeReference<>() {};
    TeamResponse response = webTestClient.get()
        .uri("/teams/{id}", team.getId())
        .exchange()
        .expectStatus().isOk()
        .expectBody(responseType)
        .returnResult()
        .getResponseBody();

    log.info("Obtained the following response {}", response);

    // Assert results
    assertNotNull(response);
    assertEquals(team.getId(), response.id());
    assertEquals(team.getName(), response.name());
    assertEquals(team.getRegion(), response.region());
    assertEquals(team.getStadium(), response.stadium());
    assertEquals(team.getFollowers(), response.followers());
}
```

Figure 8.1: Unit test - Get Team by Id

8.2 Integration Tests

After completing the development phase, integration tests were conducted to identify any issues resulting from the integration between the frontend and back-end of the platform.

Table 8.1 presents the tests carried out to the authentication module of the platform. Each User Story listed in Appendix A underwent at least one integration test, except for those that were not implemented. The Experiences module, as mentioned in Section 7.2, was not developed and, consequently, was not tested.

The integration test results comprise the following components:

- ID - The test identifier;
- US - The identifier of the User Story being tested;
- Scenario - The name of the User Story being tested;
- Description - The description of the test;
- Steps - The steps necessary to execute the test;
- Expected Result - The test's expected result;
- Result - The test's actual result;
- Status - Shows if the test Passed (P), was Not Implemented (N/I), or Failed (F).

As these tables are very extensive, the remaining are displayed in Appendix D

ID	US	Scenario	Description	Steps	Expected Result	Result	Status
1	1	Register on the platform	An unregistered user wants to register himself on the platform	- Go to registration page - Insert necessary information - Verify account in email	Account is verified successfully and the user can now log in	Account is verified successfully and the user can now log in	P
2	2	Log in to the platform	A unauthenticated user wants to login	- Go to login page - Insert login information	User is successfully logged into the platform	User is successfully logged into the platform	P
3	2	Log in to the platform	Save login information	- Go to login page - Insert login information - Select remember me checkbox	User information is saved and user is automatically logged in next time it opens the website	User information is saved and user is automatically logged in next time it opens the website	P
4	3	Recover password	A registered user wants to reset his password	- Go to login page - Press forgot password button - Insert email associated with account - Receive recovery email - Define a new password	The user can log in with the his new password	The user can log in with his new password	P
5	4	Log out	An authenticated user wants to log out of the platform	- Press the log out button	The user is logged out of the platform	The user is logged out of the platform	P

Table 8.1: Integration test results - Authentication module

8.3 Non-functional Tests

Equally as important as testing the platforms functionalities is testing their non-functional aspects, as these define how the system should operate and function. The non-functional tests, which cover all the non-functional requirements defined in Section 3.4, will be presented from the most crucial to the least.

8.3.1 Security - High Priority

Security is the top priority non-functional requirement, and it is the only requirement in this category. In Section 3.4, two scenarios were developed to address this requirement.

The first scenario involves a non-authenticated attempting to access resources or pages intended only for authenticated users. In the current version of the platform, there is only one page reserved for authenticated users - the create review page. Additionally, backend endpoints were subjected to this NFR, which made sure that only certain endpoints were available to the public. The remaining endpoints needed a valid JSON Web Token (JWT) to access them, following the microservice pattern described in Section 2.4.2.

The tests results for this scenario are presented in Table 8.2. This table is constituted by each test's ID, description, expected results, result, and status, similarly to the integration tests' table.

ID	Description	Expected Result	Result	Status
1	A unauthenticated user tries to access the create review page	The user is redirected to the homepage	The user is redirected to the homepage	P
2	Send an request to an authenticated endpoint without a JWT	Request is refused and returns a error response (401)	Request is refused and returns a error response (401)	P
3	Send an request to an authenticated endpoint with an invalid JWT	Request is refused and returns a error response (401)	Request is refused and returns a error response (401)	P
4	Send an request to an authenticated endpoint with an expired JWT	Request is refused and returns a error response (401)	Request is refused and returns a error response (401)	P
5	Send an request to an authenticated endpoint with a valid JWT	Request is accepted and returns a successful response (2xx)	Request is accepted and returns a successful response (2xx)	P

Table 8.2: Security NFR - Unauthenticated scenario test results

The second scenario addresses the issue of unauthorized access to content or information. To combat this, the backend endpoints were configured to only allow updates and deletes of a user's own content, which is verified using the access token. Additionally, retrieving information of another user does not return sensitive or personal information, such as the user's email address.

To ensure the security of the services, it is crucial to protect them from external traffic. This was achieved by implementing a CORS policy on the API Gateway, which blocked all requests originating from outside the platform. Furthermore, the football, content, and users microservices can only be accessed through the API Gateway, preventing direct external requests to these services. Moreover, the databases and S3 buckets are exclusively accessible from the mentioned microservices. The CloudFront deploy of the frontend web app has exclusive access to its own S3 bucket.

The tests conducted to ensure the correct functioning of this scenario are documented in Table 8.3, which has the same structure as Table 8.2.

ID	Description	Expected Result	Result	Status
1	A user tries to get information about another user	Sensitive information is not returned (for example, the email credential)	Sensitive information is not returned (for example, the email credential)	P
2	A user tries to update another user's information	Request is refused and returns a error response (403)	Request is refused and returns a error response (403)	P
3	A user tries to update another user's review	Request is refused and returns a error response (403)	Request is refused and returns a error response (403)	P
4	A user tries to delete another user's review	Request is refused and returns a error response (403)	Request is refused and returns a error response (403)	P
5	A user tries to update another user's comment	Request is refused and returns a error response (403)	Request is refused and returns a error response (403)	P
6	A user tries to delete another user's comment	Request is refused and returns a error response (403)	Request is refused and returns a error response (403)	P
7	A user tries to add an item to another user's collection	Request is refused and returns a error response (403)	Request is refused and returns a error response (403)	P
8	A user tries to remove an item from another user's collection	Request is refused and returns a error response (403)	Request is refused and returns a error response (403)	P
9	A user tries to delete another user's collection	Request is refused and returns a error response (403)	Request is refused and returns a error response (403)	P
10	A user tries to make a request to the backend from a non-authorized origin	Request is blocked by the API-Gateway's CORS policy	Request is blocked by the API-Gateway's CORS policy	P
11	A user tries to make a request directly to the Content Manager microservice	The request is blocked by inbound traffic rules	The request is blocked by inbound traffic rules	P
12	A user tries to make a request directly to the User Information microservice	The request is blocked by inbound traffic rules	The request is blocked by inbound traffic rules	P
13	A user tries to make a request directly to the Football Information microservice	The request is blocked by inbound traffic rules	The request is blocked by inbound traffic rules	P
14	A user tries to access the RDS database directly	The request is blocked by inbound traffic rules	The request is blocked by inbound traffic rules	P
15	A user tries to access the S3 bucket directly	The request is blocked by the bucket's permissions and policies	The request is blocked by the bucket's permissions and policies	P

Table 8.3: Security NFR - Unauthorized scenario test results

It is equally important to identify potential risks to the platform's security, which could be exploited through attacks. To do this, the tool OWASP Zed Attack Proxy (ZAP) [121] was used. This tool provides an automatic scan that identifies potential security issues present in a website, which was used on the Golden Goal platform.

The results of the scan can be found in Figure 8.2.

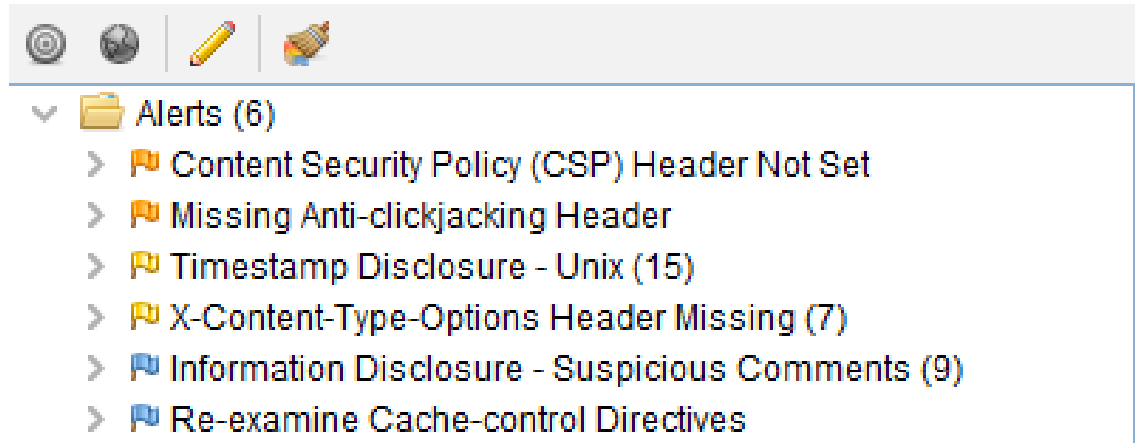


Figure 8.2: OWASP ZAP automated scan results

The security issues presented are ranked according to their severity. The first two, which are accompanied by an orange flag, correspond to the medium risk level. The following two risks have a low risk level, while the last two are marked as informational risks. There are no risks present with a high risk level.

As the tests were conducted during the final weeks of the internship, there was no time to correct the alerts found. However, rectifying these alerts should be given the utmost priority for the next stage of the platform's development.

The subsequent sections will provide a detailed account of the alerts that were discovered and propose potential solutions to address them effectively.

Content Security Policy (CSP) Header Not Set: Medium risk

Content Security Policy (CSP) is a security mechanism implemented by web applications to mitigate various types of attacks, such as cross-site scripting and data injection. Both of these have pretty severe consequences, from data theft to malware distribution [122]. CSP is able to mitigate these risks by restricting the domains from which resources can be loaded [123].

Correcting this alert is of the utmost importance, especially because Golden Goal is a platform based on user-generated content. To solve this alert, a CSP header needs to be added to the platform. This would exclude all scripts that do not belong to the Golden Goal platform.

Missing Anti-clickjacking Header: Medium risk

This alert indicates that the platform is not properly implementing protection against clickjacking attacks. These are attacks that trick users into clicking on page elements that are invisible or disguised as existing elements, which can execute unintended actions [124].

In order to solve it, the platform should include a correctly configured X-Frame-Options header or a CSP header with the "frame-ancestors" directive. Implementing these headers will help prevent attackers from tricking users by ensuring that the web page cannot be framed by malicious websites.

Timestamp Disclosure - Unix: Low risk

This vulnerability occurs when the backend includes a Unix timestamp in error messages or response headers that are returned to the client [125].

After carefully analyzing the 15 timestamps mentioned in this alert, it is apparent that this is a false positive. The timestamps are actually "nothing up my sleeve" numbers, used for initialization purposes of the SHA-256 encryption algorithm [126]. These numbers are carefully chosen to demonstrate transparency and prevent any suspicion of hidden patterns that could be exploited by attackers, ensuring the security of SHA-256 [127].

As they were represented as 32 bit unsigned integers, the automated scan mistook some of these numbers for leaked timestamps.

X-Content-Type-Options Header Missing: Low risk

This alert means that the X-Content-Type-Options header is missing from responses, which explicitly instructs the browser on how to handle the content type. This can be exploited by attackers to manipulate the responses, tricking the browser into interpreting them in an unintended manner [128].

To mitigate this alert, the responses should include the X-Content-Type-Options header with the value "nosniff". This header instructs the browser to adhere to the content type specified in the response [128].

Information Disclosure - Suspicious Comments: Informational risk

This vulnerability suggests that the responses contain potentially harmful comments that could aid attackers in gathering platform information [129].

After searching through the comments, it was found that they were all comments were displaying the copyright licenses of React, React-Router (a popular library used for navigation), and AWS Amplify (official AWS JavaScript library). Because of this, non of them revealed sensitive information nor presented a security risk.

Re-Examine Cache-control Directives: Informational Risk

Lastly, this alert indicates that due to cache-control directives not being properly configured, sensitive information may be cached. While this may be the intended approach for some files, it is important to make sure sensitive information is not cached, or it may be vulnerable to attacks [130].

The solution is to ensure that the HTTP header "cache-control" is set to "no-cache, no-store, must-revalidate" on sensitive or private information.

8.3.2 Availability - Medium Priority

As mentioned in Section 3.4, the availability of the platform is determined by two factors: the availability of the AWS resources and the code quality. Although the availability of AWS's resources is uncontrollable, the code quality plays a significant role in ensuring the overall platform's availability.

In order to ensure this, the code has been thoroughly tested, as described in Sections 8.1 and 8.2. Furthermore, the development followed the best practices for code development, such as adhering to coding standards and conducting code reviews.

While it was removed from the project's scope, the internal load balancer would benefit this NFR, as it would be a key piece for handling faults. The load balancer can monitor the health of each internal microservice and ensure that they are correctly functioning, redirecting requests only to available instances of a replicated microservice.

In conclusion, even though the internal load balancer was not developed, the applied strategies aim to reduce the number of bugs on the code and prevent possible issues that impact the overall platform availability. Although it is impossible to guarantee 100% availability, these processes aim to maximize it.

8.3.3 Performance - Medium Priority

The scenario developed for this NFR in Section 3.4 mentions that all pages should be presented in under two seconds. The following results were obtained using Chrome's built in load time, displayed on the browser's DevTools.

The load time of each page was tested ten times, with the final result being the average of all load times. All caches were cleared between tests and the times were calculated for both unauthenticated and authenticated users.

Page	Unauthenticated times	Authenticated user times
Homepage	1.09s	1.75s
Review List	1.05s	1.44s
Review	1.18s	1.62s
Review (image)	1.25s	1.73s
Review (with YouTube embed)	1.76s	2.21s
Review (with Tweet embed)	3.90s	4.41s
Create Review	-	527ms
Match List	1.04s	1.48s
Match	1.29s	1.63s
Team	1.17s	1.61s
Profile (public)	943ms	-
Profile (private)	-	1.66s
Sign in	1.12s	-

Table 8.4: Performance test results

From Table 8.4, it is evident that the majority of load times are below the two seconds threshold. The exceptions to this are the review pages that contain embedded Tweets and YouTube videos. It is worth mentioning that the extended load time for the latter is only applicable when considering authenticated users.

In this case, it is reasonable to expect that longer reviews with embedded content will take more time to load compared to regular pages. Given the variability in content types and unlimited review length, it is impractical to ensure that all reviews load within two seconds.

Furthermore, the authenticated user's load times were found to be 461ms longer in average. This is due to their requests needing to have their access tokens validated through AWS's Cognito. Additionally, their information needs to be fetched from the database and some pages become more complex when a user is logged in.

In conclusion, even though it currently meets its threshold without undergoing any improvements, it is important to keep testing this NFR. Increasing the complexity of the content present in each page may lead to a need of further optimization, either of the frontend logic or the backend structure.

8.3.4 Scalability - Low Priority

According to the scalability scenario described in Section 3.4, the system should be able to handle load spikes of at least 50% of its average load.

However, since the platform currently lacks an established user base, there is no average load to reference. Therefore, it becomes essential to examine this NFR with a future-oriented approach, to understand what amount of requests per second the system can handle and when it may be needed to be optimized.

The following tests were conducted using JMeter [131], a tool made to analyze how systems perform under load. In this setup, each user made 5 different requests over a period of one second.

To attempt to accurately simulate natural user behaviour, the following requests, which are most commonly used on the platform, were selected:

- List Users, Matches, Competitions, and Reviews;
- Get Users, Matches, and Reviews by id;
- Like Matches and Reviews;
- Create Reviews and Comments;
- Update Collections and Users.

Furthermore, the platform was loaded with all of the football information available, alongside enough users and reviews to fulfill every request. For each test, the average and maximum response times were recorded.

Number of Users	Total requests	Average	Maximum
50	250	123ms	363ms
100	500	196ms	1.18s
250	1250	623ms	3.89s
500	2500	795ms	7.51s

Table 8.5: Scalability test results

Analyzing the results, it can be seen that the average response time increases significantly from 500 to 1250 requests, as the system loses the capacity to answer such a large quantity of requests. These findings show the need for optimization measures to ensure the platform’s scalability, especially if these spikes start nearing 1250 requests per second, as seen in the third test.

The load balancer, which was removed from the internship’s scope, will help with this, as it will become possible to duplicate an instance of an overloaded microservice using Amazon’s EC2 auto scaling instances.

The databases may also need adjustments to improve their performance and scalability. Amazon RDS Read Replicas and RDS Aurora both offer horizontal scaling capabilities, allowing for data replication, which may be interesting to explore as the platform starts to grow. Vertical scalability, which upgrades the databases to more expensive versions that offer better performance, might also be a solution to consider.

8.3.5 Usability - Low Priority

Usability is crucial in a website with user-generated content. A user-friendly interface enhances the overall user experience, making it easier and more enjoyable for individuals to generate and contribute content.

To ensure that this requirement is fulfilled, usability tests were conducted with three different members of the platform's target audience. To gather diverse feedback, the candidates chosen had different ages and backgrounds. Unfortunately, due to scheduling conflicts, it was impossible to conduct tests with professional review creators, who would have provided an experienced perspective from the content creator side.

The objective of these was to ensure that users could successfully navigate, utilize, and understand the platform. With this in mind, a guideline was created, covering the great majority of the features of the website, from the registration process to searching for a match and creating a review. All participants were initially given 2 minutes to explore the website, as mentioned in this requirement's scenario. After that, the candidates followed the same guideline and were encouraged to think out loud and provide feedback.

Starting with the website navigation, all participants found it simple and user-friendly. The sidebar ensured that users never felt lost while navigating the platform. The logic of the navigation, similar to other football websites, was easy to understand and use. However, participants mentioned that the login button icon on the sidebar could be improved. They felt that it was not immediately recognizable and could be mistaken for a logout icon or an "expand sidebar" icon.

The existence of "log in to comment" or "log in to follow" buttons throughout the website helped users understand which features they could access by creating an account. To enhance this further, it would be beneficial to have the "add review" button always visible on the sidebar, as this feature came as a surprise to some users, since similar websites do not allow for user-generated content.

The review editor was found to be simple and easy to use, resembling popular text editors like Microsoft Word. Participants appreciated that it worked with familiar hotkeys for actions like undo and bold, making it fast to use. The ability to add images, YouTube videos, and Tweets to reviews was highly praised as well, as it allowed users to illustrate content and convey ideas effectively. Nonetheless, it was not clear for users that they could not save or publish a review without adding the language and expertise level details.

Regarding the reviews themselves, there are three areas for improvement. Firstly, the review's thumbnail should be visible within the review page itself, not just on the review cards. Secondly, the left side column below the author's details should not be empty. An interesting idea proposed was to include suggested reviews in that space, similar to YouTube's recommendations on a video page. This would enhance user engagement and discovery of related content. Thirdly, the review tags should also be clickable. This was one of the few requirements that were not completed in time but should be prioritized in the future, as it will allow users to quickly find reviews they are interested in.

In terms of the user profile, there are a few potential improvements. Adding a cover picture behind the username would enhance the visual appeal and personalization of the profile. Additionally, allowing users to expand the "followers" and "followed" details would simplify managing the members in these lists.

In conclusion, the usability test was a success. Every user met the metric established in the NFR's scenario, whilst providing valuable feedback to further improve the platform's design, which has been communicated to Golden Goal's design team. In the future, it would also be interesting to conduct tests with review creators, as they would provide interesting feedback from another perspective, which is equally important to consider.

Chapter 9

Conclusion

The final chapter of the document is divided into two different sections. The first one provides an overview of the entire internship, its achievements and key learnings. Lastly, Section 9.2 wraps up the document, aiming to display potential ideas, improvements, and a future direction for the Golden Goal platform.

9.1 Conclusion

The conclusion of this document marks the end of the internship, which allowed the student to integrate with the company and establish contact with the software development industry. The student applied the knowledge learned throughout the Master in Informatics Engineering, specialization in Software Engineering, by being at the core of the planning, research, development, and testing. These processes were carried out with the help of experienced members at Grama.

The first step taken in this process was to analyze the state of the art, which began with an analysis of the different types of competitors that the Golden Goal platform will face, from general web forums to platforms dedicated to sharing professional reviews. This was followed by an analysis of the different types of rating systems and how they could impact the platform. The state of the art of architecture was also analyzed, comparing monolithic and microservices architectures' advantages and disadvantages. As the chosen architecture style was a microservices architecture, research was conducted on the most commonly used microservices patterns. Backend, frontend, cloud service providers, and rich text editors were compared based on their pros and cons, leading to the choices of Spring Boot [47], React [36], AWS [5], and Lexical, respectively. Football information APIs were also researched and, due to having different advantages, both API-Football [85] and Football-data [84] were chosen for this project.

Following this, the project's functional and non-functional requirements were defined. The former were based on the User Stories created by the student and were prioritized using the MoSCoW method [90]. The latter were also given different priorities and defined using scenarios specifically created for each of them.

Subsequently, the project's architecture was created using the C4 model [6], based on the previously mentioned microservices and their patterns. Additionally, three entity-relationship diagrams were created to define the data to be kept in each microservice's database.

The second semester was based around the development of the platform, which followed the scrum-based methodology mentioned in Chapter 5. Even though the scope of the project had to be readjusted to the length of the internship, all the crucial requirements and architectural components were able to be completed and successfully tested in time. This, in addition to the negligible influence resulting from the unfinished requirements, led to the creation of a Minimum Viable Product that fulfills the company's objectives.

In conclusion, the internship provided the student with new knowledge, skills, and experiences. While partaking in the chosen methodology, the student demonstrated the ability to successfully develop and test the Golden Goal platform by exploring the technologies chosen during the first semester. Furthermore, the student researched and elaborated a structured architecture for a new project that was capable of meeting its pre-established objectives. Moreover, this architecture ended up being deployed to the AWS, which made use of its multiple cloud computing services to globally distribute the platform. All of these factors contributed to the resounding success of the internship, which brought value to the company and fostered the student's growth as a software developer.

9.2 Future Work

The future of the Golden Goal platform is very exciting. As football is such an extensive topic, there are many interesting features that could be implemented to improve the user's experience and further enrich the platform's offerings.

The first place to start would be to complete the requirements that were not completed or were removed from the scope of the internship. These are very interesting, as the Experiences bring a different type of content to the platform, and players and past seasons' matches bring more content to be reviewed. This could be accompanied by extra data and statistics about teams and players, which could be obtained from data analysis platforms such as Opta [132] and FBref [133]. As seen by the tests conducted on the non-functional requirements, certain aspects of the platform's security need to be improved. Furthermore, it may be interesting to further study the platform's usability and, in the future, improve the platform's scalability to better handle heavier loads.

The next steps to expand the platform should be the creation of a notification module, which would alert the users of interactions that happened with their content, and an administration module, to help moderate and manage the platform. Both of these have been thought of by the company, Grama, but were removed from internship before it even started, to adjust its scope to the development time available.

As a natural progression, adapting the platform for mobile devices either as a progressive web app or a dedicated mobile app would significantly increase its accessibility and reach. This adaptation would allow users to access the platform conveniently on their smartphones and "on the go".

A great suggestion that came from this internship's advisor, Prof. Pedro Abreu, is to incorporate sentiment analysis into the reviews. By analyzing the textual content of the reviews, sentiment analysis can provide insights into how a team is performing. This would fit nicely into the platform, which by its tactical and analytic nature would benefit from many fascinating statistics such as that one.

Upgrading the review text editor to include a digital tactics board would be another compelling improvement. This feature would enable users to select, position, and tactically move players on a top-down view of a football field. Users could also add arrows and drawings on top of the board to explain and display their tactical opinions and strategies. This would augment the content creation process, allowing for faster review creation and ensuring consistency and uniformity across reviews. It would also be beneficial for review creators, as the added graphics could increase the appeal of their content.

In summary, the future development of the Golden Goal platform holds immense potential for enriching the user experience. By pursuing these enhancements, the platform can continue to evolve and offer a comprehensive and engaging experience for its users.

References

- [1] Grama. Available at <https://www.grama.io/>. Last visited 11-1-2023.
- [2] Facebook. Available at <https://www.facebook.com/>. Last visited 11-1-2023.
- [3] Twitter. Available at <https://twitter.com/>. Last visited 11-1-2023.
- [4] LinkedIn. Available at <https://www.linkedin.com/feed/>. Last visited 11-1-2023.
- [5] Cloud computing services. Available at <https://aws.amazon.com>. Last visited 21-11-2022.
- [6] Simon Brown. The c4 model for visualising software architecture. Available at <https://c4model.com>. Last visited 25-11-2022.
- [7] Total football analysis. Available at <https://totalfootballanalysis.com/>. Last visited 1-10-2023.
- [8] Goalpoint. Available at <https://goalpoint.pt/>. Last visited 1-10-2023.
- [9] Lateral izquierdo. Available at <https://www.lateralesquedo.com/>. Last visited 1-10-2023.
- [10] Código futbolístico. Available at <https://codigofutbolistico.wixsite.com/codigofutbolistico>. Last visited 1-10-2023.
- [11] Startseite. Available at <https://spielverlagerung.de/>. Last visited 1-10-2023.
- [12] Reddit. Available at <https://www.reddit.com/>. Last visited 1-10-2023.
- [13] Twitter - total football analysis. Available at <https://twitter.com/totalanalysis>. Last visited 1-10-2023.
- [14] Youtube. Available at <https://www.youtube.com/>. Last visited 8-10-2023.
- [15] Imdb. Available at <https://www.imdb.com/>. Last visited 8-10-2023.
- [16] Cristina Gena, Roberto Brogi, Federica Cena, and Fabiana Venero. The impact of rating scales on user's rating behavior. In Joseph A. Konstan, Ricardo Conejo, José L. Marzo, and Nuria Oliver, editors, *User Modeling, Adaption and Personalization*, pages 123–134. Springer Berlin Heidelberg, 2011.

- [17] Tripadvisor. Available at <https://www.tripadvisor.pt/>. Last visited 8-10-2023.
- [18] E. Isaac Sparling and Shilad Sen. Rating: How difficult is it? In *Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11*, page 149–156, New York, NY, USA, 2011. Association for Computing Machinery.
- [19] Venera Tomaselli, Giulio Cantone, and Valeria Mazzeo. The polarising effect of review bomb, 04 2021.
- [20] Rotten tomatoes. Available at <https://www.rottentomatoes.com/>. Last visited 8-10-2023.
- [21] Metacritic. Available at <https://www.metacritic.com/>. Last visited 8-10-2023.
- [22] Changseung (Chang) Yoo, Warut Khern-am nuai, Jitsama Tanlamai, and Yossiri Adulyasak. Haters gonna hate? how removing downvote option impacts discussions on online forum. *SSRN*, May 2021.
- [23] IMDb Support. Why ratings are different on movie page from the top 250 list?, Oct 2018. Available at <https://community-imdb.sprinklr.com/conversations/data-issues-policy-discussions/why-ratings-are-different-on-movie-page-from-the-top-250-list/5f4a79fe8815453dba949de8>. Last visited 6-10-2022.
- [24] IMDb Support. Weighted average ratings, 2021. Available at https://help.imdb.com/article/imdb/track-movies-tv/weighted-average-ratings/GWT2DSBYVT2F25SK?ref_=helpms_helpart_inline. Last visited 6-10-2022.
- [25] Joydip Kanjilal. Pros and cons of monolithic vs. microservices architecture: Techtarget, Jan 2020. Available at <https://www.techtarget.com/searcharchitecture/tip/Pros-and-cons-of-monolithic-vs-microservices-architecture>. Last visited 25-10-2022.
- [26] Ran Ilany, Andrew Manby, JT Giri, and Chris Tozzi. Microservices vs. monoliths: Which is right for your enterprise?, Jun 2020. Available at <https://devops.com/microservices-vs-monoliths-which-is-right-for-your-enterprise/>. Last visited 25-10-2022.
- [27] Chris Richardson. Microservices pattern: Microservice architecture pattern, 2018. Available at <https://microservices.io/patterns/microservices.html>. Last visited 25-10-2022.
- [28] Chris Richardson. Microservices pattern: Database per service, 2019. Available at <https://microservices.io/patterns/data/database-per-service.html>. Last visited 25-10-2022.

-
- [29] Chris Richardson. Microservices pattern: Shared database, 2019. Available at <https://microservices.io/patterns/data/shared-database.html>. Last visited 25-10-2022.
- [30] Chris Richardson. Microservices pattern: Sagas, 2019. Available at <https://microservices.io/patterns/data/saga.html>. Last visited 25-10-2022.
- [31] Chris Richardson. Microservices pattern: Command query responsibility segregation (cqrs), 2019. Available at <https://microservices.io/patterns/data/cqrs.html>. Last visited 25-10-2022.
- [32] Chris Richardson. Microservices pattern: Api composition, 2019. Available at <https://microservices.io/patterns/security/api-composition.html>. Last visited 25-10-2022.
- [33] Chris Richardson. Microservices pattern: Access token, 2019. Available at <https://microservices.io/patterns/security/access-token.html>. Last visited 25-10-2022.
- [34] Chris Richardson. Microservices pattern: Api gateway pattern, 2019. Available at <https://microservices.io/patterns/apigateway.html>. Last visited 25-10-2022.
- [35] Sacha Greif. The state of js 2021, 2022. Available at <https://2021.stateofjs.com/en-US/libraries/front-end-frameworks>. Last visited 20-10-2022.
- [36] React – a javascript library for building user interfaces, 2013. Available at <https://reactjs.org>. Last visited 20-10-2022.
- [37] Angular, 2016. Available at <https://angular.io>. Last visited 21-10-2022.
- [38] Vue.js - the progressive javascript framework, 2014. Available at <https://vuejs.org>. Last visited 21-10-2022.
- [39] Svelte • cybernetically enhanced web apps, 2016. Available at <https://svelte.dev>. Last visited 21-10-2022.
- [40] Meta. Available at <https://www.meta.com>. Last visited 22-10-2022.
- [41] Hiren Dhaduk. 10 best frontend frameworks for web development in 2022, Sep 2022. Available at <https://www.simform.com/blog/best-frontend-frameworks>. Last visited 20-10-2022.
- [42] Jeel Patel. 10 best front end frameworks for web development in 2023, Sep 2022. Available at <https://www.monocubed.com/blog/best-front-end-frameworks>. Last visited 21-10-2022.
- [43] 10 best frontend frameworks in 2022: Technostacks, Sep 2022. Available at <https://technostacks.com/blog/best-frontend-frameworks>. Last visited 21-10-2022.
- [44] Let's build from here. Available at <https://github.com>. Last visited 21-10-2022.

- [45] Most popular backend frameworks – 2012/2022, Jan 2022. Available at <https://statisticsanddata.org/data/most-popular-backend-frameworks-2012-2022>. Last visited 22-10-2022.
- [46] Javinpaul. 10 best backend frameworks for web development in 2022, Mar 2022. Available at <https://medium.com/javarevisited/10-best-backend-frameworks-for-web-development-8d19e337f774>. Last visited 22-10-2022.
- [47] Spring boot. Available at <https://spring.io/projects/spring-boot>. Last visited 21-10-2022.
- [48] Express - node.js web application framework. Available at <https://expressjs.com>. Last visited 21-10-2022.
- [49] Django. Available at <https://www.djangoproject.com>. Last visited 21-10-2022.
- [50] Welcome to flask. Available at <https://flask.palletsprojects.com/en/2.2.x>. Last visited 21-10-2022.
- [51] Guillaume Smet. Supersonic subatomic java. Available at <https://quarkus.io>. Last visited 21-10-2022.
- [52] Alexey Krivov and Aleksandros Topalidis. Spring boot vs quarkus: Which framework to choose in 2023, Oct 2022. Available at <https://maddevs.io/blog/spring-boot-vs-quarkus>. Last visited 22-10-2022.
- [53] Doug Pierce. Quarkus vs spring boot: Which framework is right for you, Sep 2022. Available at <https://rollbar.com/blog/quarkus-vs-spring-boot>. Last visited 22-10-2022.
- [54] Nataraja Gootooru. What are the advantages and disadvantages of spring boot? Available at https://www.java2novice.com/java_interview_questions/spring-boot-pros-and-cons. Last visited 22-10-2022.
- [55] Gaetano Piazzolla. Achieve best performance: Spring-boot or node-js?, Jul 2021. Available at <https://medium.com/geekculture/achieve-best-performance-spring-boot-or-node-js-6cda6fdd9b95>. Last visited 22-10-2022.
- [56] Sophia Tondon. The pros and cons of node.js web app development: A detailed look, Mar 2022. Available at <https://javascript.plainenglish.io/the-pros-and-cons-of-node-js-web-app-development-a-detailed-look-c91a22f013c>. Last visited 22-10-2022.
- [57] Tymets Volodymyr. Express.js mobile app development: Pros and cons of node.js framework, Jul 2020. Available at <https://apiko.com/blog/express-mobile-app-development>. Last visited 22-10-2022.

-
- [58] Youssef Nader. What is django? advantages and disadvantages of using django, 2022. Available at <https://hackr.io/blog/what-is-django-advantages-and-disadvantages-of-using-django>. Last visited 22-10-2022.
- [59] Matthew Deery. 9 pros and cons of the django framework: A coder's guide, Aug 2021. Available at <https://careerfoundry.com/en/blog/web-development/django-framework-guide>. Last visited 22-10-2022.
- [60] DataFlair Team. Django advantages and disadvantages - why you should choose django?, Jun 2021. Available at <https://data-flair.training/blogs/django-advantages-and-disadvantages>. Last visited 22-10-2022.
- [61] Matthew Deery. The flask web framework: A beginner's guide, Aug 2022. Available at <https://careerfoundry.com/en/blog/web-development/what-is-flask>. Last visited 22-10-2022.
- [62] Microsoft azure. Available at <https://azure.microsoft.com/pt-pt>. Last visited 13-10-2022.
- [63] Google cloud. Available at <https://cloud.google.com>. Last visited 13-10-2022.
- [64] Felix Richter. Infographic: Amazon, microsoft & google dominate cloud market, 2022. Available at <https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers>. Last visited 22-10-2022.
- [65] Andriy Stashko. Top cloud service providers: A quick comparison, Dec 2021. Available at <https://www.avenga.com/magazine/top-cloud-service-providers>. Last visited 22-10-2022.
- [66] Top 5 cloud service providers: A comparison - recro: Blog, Nov 2019. Available at <https://recro.io/blog/top-5-cloud-service-providers>. Last visited 22-10-2022.
- [67] Firebase. Available at <https://firebase.google.com>. Last visited 13-10-2022.
- [68] Jeferson Mari. Jefmari/awesome-wysiwyg: A curated list of awesome wysiwyg editors., Apr 2017. Available at <https://github.com/JefMari/awesome-wysiwyg#for-react>. Last visited 22-10-2022.
- [69] 5 best open-source wysiwyg editors for react (2023), 2022. Available at <https://www.kindacode.com/article/popular-open-source-wysiwyg-editors-for-react>. Last visited 22-10-2022.
- [70] Yajas Sri Wickramasinghe. Top 5 rich text editors for react in 2021, Aug 2021. Available at <https://blog.bitsrc.io/top-5-rich-text-editors-for-react-in-2021-628fecf0f7e0>. Last visited 22-10-2022.

- [71] Florian Kruger. Switching rich text editors, part 1: Picking tiptap, Feb 2022. Available at <https://www.ashbyhq.com/blog/engineering/tiptap-part-1>. Last visited 22-10-2022.
- [72] Draft.js. Available at <https://draftjs.org>. Last visited 22-10-2022.
- [73] Lexical. Available at <https://lexical.dev>. Last visited 22-10-2022.
- [74] trueadm. Internally, we've been replacing draft.js with lexical. to be clear though, they...: Hacker news, Apr 2022. Available at <https://news.ycombinator.com/item?id=31022152>. Last visited 22-10-2022.
- [75] Jpuri. react-draft-wysiwyg. Available at <https://github.com/jpuri/react-draft-wysiwyg>. Last visited 22-10-2022.
- [76] Quill. Available at <https://quilljs.com>. Last visited 22-10-2022.
- [77] Zenoamaro. React-quill. Available at <https://github.com/zenoamaro/react-quill>. Last visited 22-10-2022.
- [78] Slatejs. Available at <https://www.slatejs.org>. Last visited 22-10-2022.
- [79] Tiptap. Available at <https://tiptap.dev>. Last visited 22-10-2022.
- [80] Mayowa Ojo. Building a rich text editor with lexical and react, May 2022. Available at <https://blog.logrocket.com/build-rich-text-editor-lexical-react>. Last visited 22-10-2022.
- [81] Api hub - free public & open rest apis. Available at <https://rapidapi.com/hub>. Last visited 15-10-2022.
- [82] Joe Kampschmidt. Guide to football/soccer data and apis, Dec 2019. Available at <https://www.jokecamp.com/blog/guide-to-football-and-soccer-data-and-apis>. Last visited 15-10-2022.
- [83] Public-apis/public-apis: A collective list of free apis, Jul 2022. Available at <https://github.com/public-apis/public-apis#sports--fitness>. Last visited 15-10-2022.
- [84] The dev-friendly football api. Available at <https://www.football-data.org>. Last visited 15-10-2022.
- [85] Football - restful api for football data. Available at <https://www.api-football.com>. Last visited 15-10-2022.
- [86] Mxa Rehkopf. User stories: Examples and template. Available at <https://www.atlassian.com/agile/project-management/user-stories>. Last visited 11-10-2022.
- [87] Max Rehkopf. Agile epics: definition, examples, and templates. Available at <https://www.atlassian.com/agile/project-management/epics>. Last visited 11-10-2022.

-
- [88] Matthew Martin. What is a functional requirement in software engineering?, 2022. Available at <https://www.guru99.com/functional-requirement-specification-example.html>. Last visited 11-10-2022.
- [89] Ron Yang. User stories vs. requirements, Jan 2018. Available at <https://www.aha.io/blog/user-stories-vs-product-requirements>. Last visited 11-10-2022.
- [90] Moscow prioritization, 2022. Available at <https://www.productplan.com/glossary/moscow-prioritization>. Last visited 11-10-2022.
- [91] Dean Leffingwell. Nonfunctional requirements, Aug 2021. Available at <https://www.scaledagileframework.com/nonfunctional-requirements>. Last visited 11-10-2022.
- [92] Non-functional requirements: Examples, types, how to approach, Feb 2020. Available at <https://www.altexsoft.com/blog/non-functional-requirements>. Last visited 12-10-2022.
- [93] Len Bass, Paul Clements, and Rick Kazman. *Software architecture in practice*. Addison-Wesley, 2022. Available at . Last visited 12-10-2022.
- [94] Available at <https://www.cs.unb.ca/~wdu/cs6075w10/sa2.htm>. Last visited 12-10-2022.
- [95] Alina Nechvolod. 12 software quality attributes to launch a high-quality product, Jul 2022. Available at <https://syndicode.com/blog/12-software-architecture-quality-attributes>. Last visited 12-10-2022.
- [96] Everything you need to know about gdpr compliance, Apr 2019. Available at <https://gdpr.eu/compliance>. Last visited 12-10-2022.
- [97] Aws service level agreements (slas). Available at <https://aws.amazon.com/legal/service-level-agreements/>. Last visited 13-10-2022.
- [98] Nikolay Ashanin. Quality attributes in software architecture, Jun 2019. Available at <https://medium.com/@nvashanin/quality-attributes-in-software-architecture-3844ea482732>. Last visited 13-10-2022.
- [99] Kristen Baker. 11 website page load time statistics you need [+ how to increase conversion rate], Jun 2022. Available at <https://blog.hubspot.com/marketing/page-load-time-conversion-rates>. Last visited 13-10-2022.
- [100] What is stress testing, Jun 2022. Available at <https://k6.io/docs/test-types/stress-testing/>. Last visited 27-06-2023.
- [101] Usability: A part of the user experience, Aug 2021. Available at <https://www.interaction-design.org/literature/article/usability-a-part-of-the-user-experience>. Last visited 13-10-2022.

- [102] Amazon cognito. Available at <https://aws.amazon.com/cognito>. Last visited 21-11-2022.
- [103] Muhammad A. Kazi. Performance analysis of content delivery network (cloudfront) based on amazon infrastructure, 2012. Available at <https://aws.amazon.com/cloudfront>. Last visited 21-11-2022.
- [104] Load balancer - amazon elastic load balancer (elb). Available at <https://aws.amazon.com/elasticloadbalancing>. Last visited 21-11-2022.
- [105] Donald J. Daly. Secure and resizeable cloud computing. Available at <https://aws.amazon.com/ec2>. Last visited 21-11-2022.
- [106] Dietmar Kopitz and Bev Marks. Fully managed relational database. Available at <https://aws.amazon.com/rds>. Last visited 21-11-2022.
- [107] Cloud object storage. Available at <https://aws.amazon.com/s3>. Last visited 21-11-2022.
- [108] Sarang Shaikh and Sindhu Abro. Comparison of traditional & agile software development methodology: A short survey. In *International Journal of Software Engineering and Computer Systems*, 2019.
- [109] Diogo Espírito Santo. Top 5 main agile methodologies: Advantages and disadvantages, Mar 2022. Available at <https://www.xpand-it.com/blog/top-5-agile-methodologies>. Last visited 30-10-2022.
- [110] Martin Fowler and Jim Highsmith. The agile manifesto, 2000.
- [111] Ken Schwaber and Jeff Sutherland. The scrum guide, 2012.
- [112] Ken Schwaber. The scrum framework poster, 2020. Available at <https://www.scrum.org/resources/scrum-framework-poster>. Last visited 30-10-2022.
- [113] Gantt.com. Available at <https://www.gantt.com>. Last visited 13-10-2022.
- [114] Free online gantt chart maker that's easy to use. Available at <https://www.teamgantt.com>. Last visited 9-1-2023.
- [115] Project Management Institute. *A guide to the Project Management Body of Knowledge: (PMBOK Guide)*. Project Management Institute, 2017.
- [116] Harold Kerzner. *Project Management: A systems approach to planning, scheduling, and controlling*. John Wiley & Sons, Inc., 2022.
- [117] Duncan Jones. Risk: Identification, assessment and contingency planning, May 2016. Available at <https://hexagoninnovating.weebly.com/current-blogs/risk-identification-assessment-and-contingency-planning>. Last visited 13-10-2022.
- [118] Linear: a better way to build products, 2022. Available at <https://linear.app/>. Last visited 1-7-2023.

-
- [119] Available at <https://junit.org/>. Last visited 2-7-2023.
- [120] Available at <https://www.javatpoint.com/control-flow-testing-in-white-box-testing/>. Last visited 2-7-2023.
- [121] Available at <https://www.zaproxy.org/>. Last visited 3-7-2023.
- [122] MozDevNet. Content security policy (csp) - http: Mdn. Available at <https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>. Last visited 3-7-2023.
- [123] Nov 2020. Available at <https://www.imperva.com/learn/application-security/content-security-policy-csp-header/#:~:text=Mitigating%20Cross%20Site%20Scripting,the%20source%20of%20the%20content..> Last visited 3-7-2023.
- [124] Available at <https://portswigger.net/web-security/clickjacking>. Last visited 3-7-2023.
- [125] Available at <https://www.zaproxy.org/docs/alerts/10096/>. Last visited 3-7-2023.
- [126] Available at <https://rst.im/magic/sha256>. Last visited 3-7-2023.
- [127] David Wong. Nothing up my sleeve numbers posted february 2015. Available at <https://www.cryptologie.net/article/221/nothing-up-my-sleeve-numbers/>. Last visited 3-7-2023.
- [128] Available at <https://www.iothreat.com/blog/x-content-type-options-header-missing#:~:text=The%20%2DContent%2DType,perform%20content%2Dtype%20sniffing%20attacks..> Last visited 3-7-2023.
- [129] Available at <https://www.zaproxy.org/docs/alerts/10027/>. Last visited 3-7-2023.
- [130] Available at <https://www.zaproxy.org/docs/alerts/10015/>. Last visited 3-7-2023.
- [131] Available at <https://jmeter.apache.org/>. Last visited 3-7-2023.
- [132] Apr 2023. Available at <https://www.statsperform.com/opta/>. Last visited 1-7-2023.
- [133] Available at <https://fbref.com/en/>. Last visited 1-7-2023.

Appendices

Appendix A

User stories

Epic-1: Authentication

US-1: Register

- (a) **As an unauthenticated user, I want to register myself on the platform so that I can log in to access its functionalities.**

I want to insert:

- Name
- Position
- Username
- Email
- Password
- Password confirmation

US-2: Log in

- (a) **As a registered user, I want to be able to log in to the platform so that I can access its functionalities.**

I want to insert:

- Username*
- Password* (* mandatory field)

(I want to save my login information so that I can log in faster next time.)

US-3: Recover password

- (a) **As a registered user, I want to be able to recover my forgotten password so that I can log in again to the platform.**

(I want to insert my email address and receive a recovery email. It is mandatory to define a new password to access my account.)

US-4: Log out

- (a) **As a logged-in user, I want to log out of the platform so that my personal details are no longer accessible on the current browser.**

Epic-2: Personal Profile

US-5: Personal information

- (a) **As an** authenticated user, **I want to** be able to access all of my personal information **so that** I can gather any information necessary.

I want to see:

- Name
- Position
- Username
- Email address
- Profile picture
- Collections

US-6: Change name

- (a) **As an** authenticated user, **I want to** change my name **so that** I can correct or update it as needed.

US-7: Change position

- (a) **As an** authenticated user, **I want to** change my position **so that** I can correct or update it as needed.

US-8: Change username

- (a) **As an** authenticated user, **I want to** change my username **so that** I can correct or update it as needed.

US-9: Change email address

- (a) **As an** authenticated user, **I want to** change my email **so that** I can correct or update it as needed.

US-10: Change password

- (a) **As an** authenticated user, **I want to** change my password **so that** I can keep my account safe and updated.

US-11: Change profile picture

- (a) **As an** authenticated user, **I want to** change my profile picture **so that** I can update it or remove it as needed.

US-12: See collections

- (a) **As an** authenticated user, **I want to** see my collections **so that** I can consult and change them as needed.

US-13: Manage Collections

- (a) **As an** authenticated user, **I want to** create a new collection **so that** I can use them to store content as needed.

I want to choose:

- Collection name
 - Collection visibility level - private or public
 - Collection content type - Matches or reviews
- (b) **As an** authenticated user, **I want to** delete a collection **so that** I can remove unwanted collections from my account, keeping it updated and organized.
- (c) **As an** authenticated user, **I want to** rename a collection **so that** I can keep my collections updated and organized.
- (d) **As an** authenticated user, **I want to** change a collection's visibility **so that** I can decide which collections to show to the public.

US-14: Delete items from collections

- (a) **As an** authenticated user, **I want to** delete items from a collection **so that** I can keep the collection updated and organized.

US-15: See reviews and drafts

- (a) **As an** authenticated user, **I want to** see my reviews and drafts **so that** I can edit and delete them.

US-16: Delete reviews and drafts

- (a) **As an** authenticated user, **I want to** delete my reviews and drafts **so that** I can remove unwanted content from my account.

US-17: See reviews' statistics

- (a) **As an** authenticated user, **I want to** be able to see statistics from my reviews **so that** I gather information from them.

I want to see:

- Number of likes
- Number of views
- Number of shares

US-18: See experiences and drafts

- (a) **As an** authenticated user, **I want to** see my experiences and drafts **so that** I can edit and delete them.

US-19: Delete experiences and drafts

- (a) **As an** authenticated user, **I want to** delete my experiences and drafts **so that** I can remove unwanted content from my account.

US-20: See experiences' statistics

- (a) **As an** authenticated user, **I want to** be able to see statistics from my experiences **so that** I gather information from them.

US-21: See personal statistics

- (a) **As an** authenticated user, **I want to** see all of my personal statistics, **so that** I can gather any information necessary about my website activity.

I want to see:

- Frequent content level
- Frequent tags
- Joining date
- Total likes
- Total comments
- Total shares
- Number of experiences published
- Number of reviews published

Epic-3: Public Profile

US-22: See an author's personal information / Profile

- (a) **As a user, I want to** see an author's personal information **so that** I can gather any information necessary about the author.

I want to see:

- Name
- Profile picture
- Position

US-23: See author's statistics

- (a) **As a user, I want to** see an author's statistics **so that** I can gather any information necessary about the author's website activity.

I want to see:

- Frequent content level
- Frequent tags
- Joining date
- Total likes
- Total comments
- Total shares
- Number of followers
- Number of followed users

US-24: Follow the author

- (a) **As an authenticated user, I want to** follow an author **so that** I can keep up with his new content releases.

US-25: See author's reviews

- (a) **As a user, I want to** see an author's reviews **so that** I can search them and read the ones I find most interesting.

I want to see:

- Title
 - Content level
 - Comment count
 - Rating
 - Associated match
- (b) **As a user, I want to filter** an author's review list **so that** I can search for a specific review more easily.
I want to filter by:
- Language
 - Content level
 - Popularity
 - Teams
 - Coaches
- (c) **As a user, I want to sort** an author's review list **so that** I can search for a specific review more easily.
I want to sort by:
- Date
 - Popularity

US-26: See the author's experiences

- (a) **As a user, I want to see** an author's experiences **so that** I can search and read the ones I find most interesting.
I want to see:
- Title
 - Description
 - Comment count
 - Rating
 - Associated match
- (b) **As a user, I want to filter** an author's experience list **so that** I can search for a specific experience more easily.
I want to filter by:
- Rating
 - Popularity
- (c) **As a user, I want to sort** an author's experience list **so that** I can search for a specific experience more easily.
I want to sort by:
- Date
 - Popularity

Epic-4: Reviews

US-27: See all reviews

- (a) **As a user, I want to** see all reviews **so that** I can search for interesting reviews to read.

I want to see:

- Title
- Author
- Content level
- Associated match
- Thumbnail
- Tags

- (b) **As a user, I want to** filter the reviews **so that** I can search for a specific review more easily.

I want to filter by:

- Language
- Content level
- Popularity
- Teams
- Coaches

- (c) **As a user, I want to** search for reviews using text **so that** I can search for a specific review more easily.

I want to search for:

- Title text
- Review text
- Tags
- Teams
- Coaches
- Competitions
- Players

- (d) **As a user, I want to** sort the reviews **so that** I can search for specific reviews more easily.

I want to sort by:

- Date
- Popularity

US-28: See trending reviews

- (a) **As an unauthenticated user, I want to** see the trending reviews **so that** I can find interesting reviews more easily.

US-29: See recent reviews by followed authors

- (a) **As an** authenticated user, **I want to** see the most recent reviews written by authors I follow **so that** I can keep up with their new reviews.

US-30: See popular authors

- (a) **As a** user, **I want to** see the most popular authors **so that** I can find interesting reviews more easily.

US-31: Navigate through tags

- (a) **As a** user, **I want to** navigate through review tags **so that** I can find similar reviews more easily.

US-32: See a full review

- (a) **As a** user, **I want to** read the full text of a review **so that** I can learn about an author's opinion on a match.

I want to see:

- Title
- Body
- Images

- (b) **As a** user, **I want to** see the match associated with the review I am reading **so that** I can understand which match the review is discussing.

I want to see:

- Teams
- Score
- Competition
- Date
- Coaches

- (c) **As a** user, **I want to** see details of the review I am reading **so that** I can gather more information about it.

I want to see:

- Author:
 1. Name
 2. Position
 3. Number of reviews published
 4. Joining date
- Content level
- Publishing date
- Number of likes
- Number of comments

US-33: Share on social media

- (a) **As a** user, **I want to** share a review on social media **so that** my followers can know about the review.

US-34: Comment review

- (a) **As an** authenticated user, **I want to** comment on a review **so that** I can share my opinion about the review I am reading.
- (b) **As an** authenticated user, **I want to** edit a comment **so that** I can correct any mistakes made.
- (c) **As an** authenticated user, **I want to** delete a comment **so that** I can remove unwanted comments from my account.

US-35: Like the review

- (a) **As an** authenticated user, **I want to** like a review **so that** I can show my appreciation of the review.

US-36: Follow the author

- (a) **As an** authenticated user, **I want to** follow the review's author **so that** I can keep up with his new content.

US-37: Add the review to a collection

- (a) **As an** authenticated user, **I want to** add the review to a chosen collection **so that** I can keep my favorite reviews organized.

US-38: Create a review

- (a) **As an** authenticated user, **I want to** create a review **so that** I can write my opinion on a match.
- (b) **As an** authenticated user, **I want to** edit my reviews' content **so that** I can add, change, or remove content from them.

I want to be able to:

- Add and change the title
- Use a text editor to add, change, and remove body text
- Use a text editor to position images alongside the text
- Use an image gallery to quickly add, change, and delete images and captions
- Add and change the thumbnail

- (c) **As an** authenticated user, **I want to** associate a match with my review **so that** the readers can understand which match I am discussing.

I want to be able to:

- See a list of matches
- Filter list of matches by home/away and by date
- Select the match to associate

- (d) **As an** authenticated user, **I want to** define my review's details **so that** readers can easily search and understand what the review is.

I want to be able to:

- Change review language
- Change the review's content level

- Add and remove tags
- (e) **As an** authenticated user, **I want to** preview my review **so that** I can prevent errors from being published.
- (f) **As an** authenticated user, **I want to** publish my review **so that** other people can read it.
- (g) **As an** authenticated user, **I want to** save my review as a draft **so that** I can continue writing it at another time.

Epic-5: Experiences

US-39: See all experiences

- (a) **As a** user, **I want to** see all experiences **so that** I can search for interesting experiences to read.
I want to see:
 - Title
 - Author
 - Rating
 - Associated match
 - Thumbnail
- (b) **As a** user, **I want to** filter the experiences **so that** I can search for a specific experience more easily.
I want to filter by:
 - Rating
 - Popularity
- (c) **As a** user, **I want to** search for experiences using text **so that** I can search for a specific experience more easily.
I want to search for:
 - Title text
 - Experience text
 - Tags
 - Teams
 - Competitions
- (d) **As a** user, **I want to** sort the experiences **so that** I can search for specific experiences more easily.
I want to sort by:
 - Date
 - Popularity

US-40: See trending experiences

- (a) **As an** unauthenticated user, **I want to** see the trending experiences **so that** I can find interesting experiences more easily.

US-41: See recent experiences by followed authors

- (a) **As an** authenticated user, **I want to** see the most recent experiences written by authors I follow **so that** I can keep up with their new experiences.

US-42: See a full experience

- (a) **As a** user, **I want to** read the full description of an experience **so that** I can learn about an author's experience in a live match.

I want to see:

- Title
- Rating
- Description
- Images

- (b) **As a** user, **I want to** see the match associated with the experience I am reading **so that** I can understand which match the experience is discussing.

I want to see:

- Teams
- Score
- Competition
- Date
- Coaches

- (c) **As a** user, **I want to** see details of the experience I am reading **so that** I can gather more information about it.

I want to see:

- Author:
 1. Name
 2. Position
 3. Number of reviews published
 4. Number of experiences published
 5. Joining date
- Content level
- Publishing date
- Number of likes
- Number of comments

US-43: Share on social media

- (a) **As a** user, **I want to** share an experience on social media **so that** I can my followers can know about the experience.

US-44: Comment experience

- (a) **As an** authenticated user, **I want to** comment on an experience **so that** I can share my opinion about the experience I am reading.

US-45: Follow the author

- (a) **As an** authenticated user, **I want to** follow the experience's author **so that** I can keep up with his new content.

US-46: Create an experience

- (a) **As an** authenticated user, **I want to** create an experience **so that** I can share my personal experience on a particular live match.
- (b) **As an** authenticated user, **I want to** edit my reviews' and drafts' content **so that** I can add, change, or remove content from them.
I want to be able to:
- Add and change the title
 - Use a text editor to add, change, and remove body text
 - Use an image gallery to quickly add and delete images and captions
 - Add and change the thumbnail
 - Add and change the rating
- (c) **As an** authenticated user, **I want to** associate a match with my experience **so that** the readers can understand which match I am discussing.
I want to be able to:
- See a list of matches
 - Filter list of matches by home/away and by date
 - Select the match to associate
- (d) **As an** authenticated user, **I want to** preview my experience **so that** I can prevent errors from being published.
- (e) **As an** authenticated user, **I want to** publish my experience **so that** other people can read it.
- (f) **As an** authenticated user, **I want to** save my experience as a draft **so that** I can continue writing it at another time.

Epic-6: Matches

US-47: See all matches

- (a) **As a** user, **I want to** see all matches **so that** I can search for interesting matches to learn about.
I want to see:
- Teams
 - Score
 - Location
 - Date
 - Competition
- (b) **As a** user, **I want to** see all matches grouped **so that** it is easier to identify each match.
I want to see the matches grouped by:
- Date

- Competition

(c) **As a user, I want to** filter the matches **so that** I can search for a specific match more easily.

I want to filter by:

- Competition
- Matchday
- Team
- Coach

(d) **As a user, I want to** search for matches using text **so that** I can search for a specific review more easily.

I want to search for:

- Teams
- Coaches
- Competitions

US-48: See match information

(a) **As a user, I want to** see a match's information **so that** I can gather information about it.

I want to see:

- Location
- Competition
- Season
- Coaches

US-49: See match details

(a) **As a user, I want to** see a match's details **so that** I can gather detailed information about it.

I want to see:

- Goals
- Statistics
- Lineups
- Most valuable player (MVP)

US-50: Mark match as watched

(a) **As an** authenticated user, **I want to** mark the match as watched **so that** I can keep my watched matches organized.

US-51: Add match to collection

(a) **As an** authenticated user, **I want to** add the match to a chosen collection **so that** I can keep my favorite matches organized.

US-52: Like match

- (a) **As an** authenticated user, **I want to** like a match **so that** I can share my opinion of the match.

US-53: Follow team

- (a) **As an** authenticated user, **I want to** follow a team **so that** I can keep up with new scores and content related to the team.

US-54: See related reviews

- (a) **As a** user, **I want to** see reviews related to that match **so that** I can learn about other people's opinions about it.

I want to see:

- Title
- Author
- Content level
- Tags

- (b) **As a** user, **I want to** see the most popular review related to that match **so that** I can easily find an interesting review about it.

I want to see:

- Title
- Author
- Content level
- Tags

- (c) **As a** user, **I want to** filter the reviews **so that** I can search for a specific review more easily.

I want to filter by:

- Tags
- Language
- Content level
- Popularity

US-55 : See related experiences

- (a) **As a** user, **I want to** see experiences related to that match **so that** I can learn about other people's live experience on that match.

I want to see:

- Title
- Author
- Rating
- Thumbnail

- (b) **As a** user, **I want to** see the most popular experience related to that match **so that** I can easily find interesting live experiences related to that match's live experience.

I want to see:

- Title
- Author
- Rating
- Thumbnail

(c) **As a user, I want to filter the experiences so that** I can search for a specific experience more easily.

I want to filter by:

- Popularity

Epic-7: Teams

US-56: See team information

(a) **As a user, I want to see a team's information so that** I can gather information about it.

I want to see:

- Name
- Crest
- Coach
- Stadium
- Foundation date

US-57: See a team's matches

(a) **As a user, I want to see a team's matches so that** I can gather information about their scores and search content about a specific match.

I want to see:

- Teams
- Score
- Review count
- Experience count
- Competition

(b) **As a user, I want to see all matches grouped so that** it is easier to identify each match.

I want to see the matches grouped by:

- Competition)

(c) **As a user, I want to filter the matches so that** I can search for a specific match more easily.

I want to filter by:

- Season

US-58: See a team's statistics

- (a) **As a user, I want to** see a team's statistics **so that** I can gather information about their performances.

I want to see:

- Matches played
- Goals scored
- Goals conceded
- Number of wins
- Number of ties
- Number of losses
- Number of clean sheets

- (b) **As a user, I want to** filter the statistics **so that** I can see their evolution and gather more information about the team.

I want to filter by:

- Season

- (c) **As a user, I want to** select a statistic and see the matches that make it up **so that** I can gather more information about the team.

I want to see

- Teams
- Score
- Competition
- Number of reviews
- Number of experiences

- (d) **As a user, I want to** see the matches grouped **so that** it is easier to identify each match.

I want to see the matches grouped by:

- Competition

US-59: See related reviews

- (a) **As a user, I want to** see reviews related to the team **so that** I can learn about other people's opinions about them.

I want to see

- Title
- Author
- Content level
- Tags
- Associated match

- (b) **As a user, I want to** filter the reviews **so that** I can search for a specific review more easily.

I want to filter by:

- Season

- Language
- Content level
- Popularity
- Coaches

US-60: See related experiences

- (a) **As a user, I want to** see experiences related to that team **so that** I can learn about other people's opinions about their live experiences.

I want to see:

- Title
- Author
- Rating
- Associated match

- (b) **As a user, I want to** filter the experiences **so that** I can search for a specific experience more easily.

I want to filter by:

- Season
- Popularity

US-61: Follow team

- (a) **As an authenticated user, I want to** follow a team **so that** I can keep up with new scores and content related to the team.

Epic-8: Homepage

US-62: See reviews

- (a) **As a user, I want to** see the most popular reviews **so that** I can easily find interesting reviews to read.

I want to see:

- Title
- Match
- Author
- Date
- Tags
- Content level
- Thumbnail

- (b) **As a user, I want to** see the most recent reviews **so that** I can easily find new reviews to read.

I want to see:

- Title
- Match

-
- Author
 - Date
 - Tags
 - Content level
- (c) **As an** authenticated user, **I want to** see reviews from followed authors **so that** I can keep up to date with their new reviews.
I want to see:
- Title
 - Match
 - Author
 - Date
 - Tags
 - Content level

US-63: See experiences

- (a) **As a** user, **I want to** see the most popular experiences **so that** I can easily find interesting experiences to read about.
I want to see:
- Title
 - Match
 - Author
 - Date
- (b) **As a** user, **I want to** see the most recent experiences **so that** I can easily find new experiences to read about.
I want to see:
- Title
 - Match
 - Author
 - Date
- (c) **As an** authenticated user, **I want to** see experiences from followed authors **so that** I can keep up to date with their new experiences.
I want to see:
- Title
 - Match
 - Author
 - Date
- (d) **As an** authenticated user, **I want to** see experiences from followed teams **so that** I can find more experiences about my favorite teams.
I want to see:
- Title

- Match
- Author
- Date

US-64: See matches

(a) **As a user, I want to** see the most popular matches **so that** I can easily find interesting matches to look into.

I want to see:

- Teams
- Score
- Date
- Competition
- Coaches
- Number of reviews
- Number of experiences
- Number of likes

(b) **As a user, I want to** see the most recent matches **so that** I can easily find new matches to look into.

I want to see:

- Teams
- Score
- Date
- Competition
- Number of reviews
- Number of experiences

(c) **As an authenticated user, I want to** see matches from followed teams **so that** I can keep up to date with my favorite teams

I want to see:

- Teams
- Score
- Date
- Competition
- Number of reviews
- Number of experiences

US-65: See teams

(a) **As an authenticated user, I want to** see my followed teams **so that** I can keep up to date with their matches, reviews, and experiences.

I want to see:

- Name

- Crest
- Number of matches
- Number of reviews
- Number of experiences

US-66: Search Menu

(a) **As a user, I want to** search using text **so that** I can easily discover new content.

I want to search for:

- Reviews
- Experiences
- teams
- Tags

(b) **As a user, I want to** filter the search results **so that** I can easily find the specific content I am looking for.

I want to filter by:

- Type
- Popularity
- Language

(c) **As a user, I want to** sort the search results **so that** I can easily find the specific content I am looking for.

I want to sort by:

- Date
- Popularity

Appendix B

Functional Requirements

Area	Id	Requirements	Logged Off User	Logged In User	Priority
Authentication	FR-1	Login	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Must
	FR-2	Logout	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-3	Register	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Must
	FR-4	Recover password	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Must
	FR-5	Change password	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-6	Remember login information	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Could

Table B.1: Authentication functional requirements

Area	Id	Requirements	Logged Off User	Logged In User	Priority
Profile	FR-7	See and edit your profile picture	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-8	See and edit personal information	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-9	Follow an author	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-10	See frequent content level	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-11	See frequent tags	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-12	See joining date	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-13	See total likes, comments and shares	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-14	See number of experiences shared	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-15	See number of reviews published	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-16	See notification updates	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Won't
Collection management	FR-17	Create collections	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-18	Delete collections	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-19	Add collection items	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-20	Remove collection items	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-21	See collection items	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-22	Set and change collection name	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-23	Set and change collection visibility	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
Collection list	FR-24	See author's collections	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-25	Filter collection list	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Should

Table B.2: Profile functional requirements

Appendix B

Area	Id	Requirements	Logged Off User	Logged In User	Priority
List Reviews	FR-26	List all reviews	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-27	Search review list by text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-28	Sort review list	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-29	Apply filters to review list	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-30	Show trending reviews	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could
Review	FR-31	Read full review	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-32	See associated match	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-33	See review statistics	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-34	Filter review statistics	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-35	See a specific review's details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-36	Share on social media	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-37	Comment review	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-38	Like the review	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-39	Listen to review	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Won't
Tags	FR-40	List related tags	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could
Add review	FR-41	Choose match to review	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-42	Change associated match	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-43	Add a title	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-44	Add a body	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-45	Add, delete and replace images	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-46	See image gallery	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-47	Add thumbnail	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-48	Add language	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-49	Add content level	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-50	Add tags	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-51	Preview reievw	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-52	Publish review	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-53	Save draft	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	Experience list	FR-54	List all experiences	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
FR-55		Filter experiences	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Should
FR-56		Sort experience	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could
FR-57		Search experience by text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Should
FR-58		Show trending experiences	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could
Experience	FR-59	Read full experience	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-60	See associated match	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-61	See a specific experience's details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-62	Share on social media	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-63	Comment experience	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
Add experience	FR-64	Associate experience with match	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-65	Change associated match	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-66	Add title	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-67	Add body	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-68	Add, delete and replace images	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-69	See image gallery	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-70	Add language	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-71	Add thumbnail	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-72	Preview experience	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-73	Publish experience	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-74	Save draft	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should

Table B.3: Content functional requirements

Area	Id	Requirements	Logged Off User	Logged In User	Priority
List matches	FR-75	See matches	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-76	Search matches by text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-77	Apply filters to match list	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Should
Match	FR-78	See match information	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-79	See match details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-80	Mark match as watched	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Should
	FR-81	Like match	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
Team	FR-82	See team information	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-83	Follow team	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Must
	FR-84	See team statistics	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-85	See statistics details	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could
	FR-86	Filter statistics by season	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Could

Table B.4: Football functional requirements

Appendix C

Functionality Showcase

Authentication Module

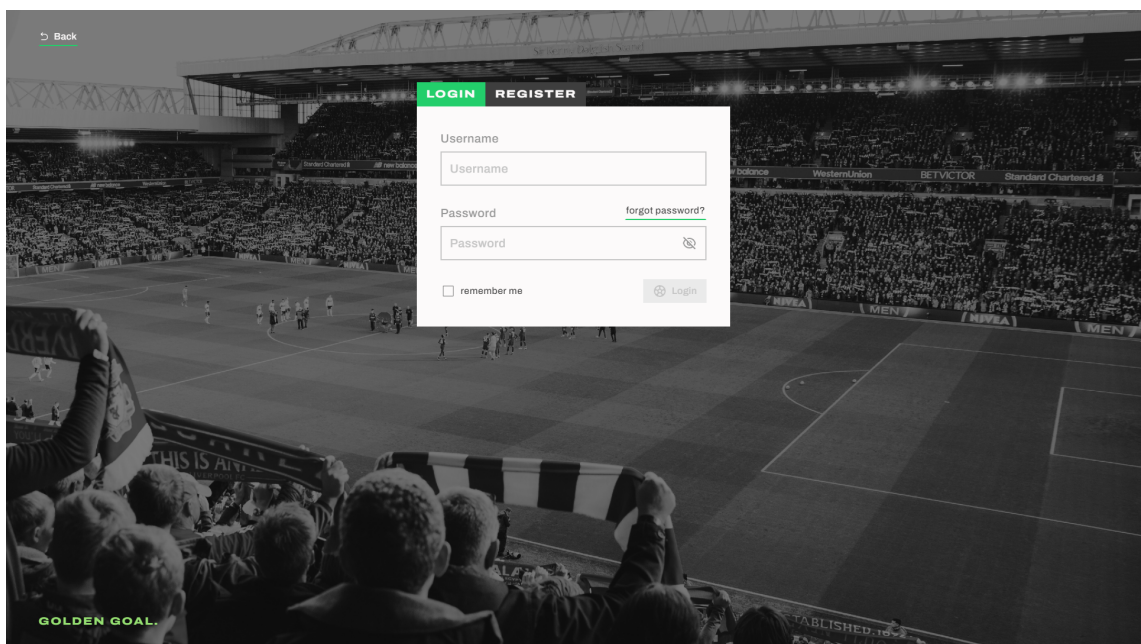


Figure C.1: Login page

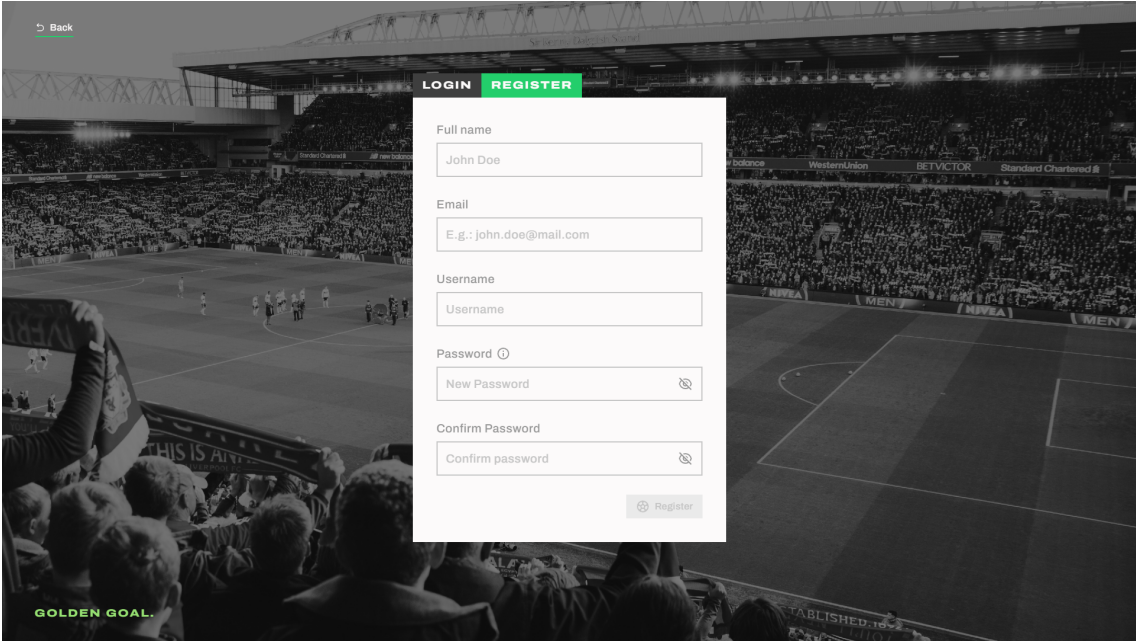


Figure C.2: Registration page

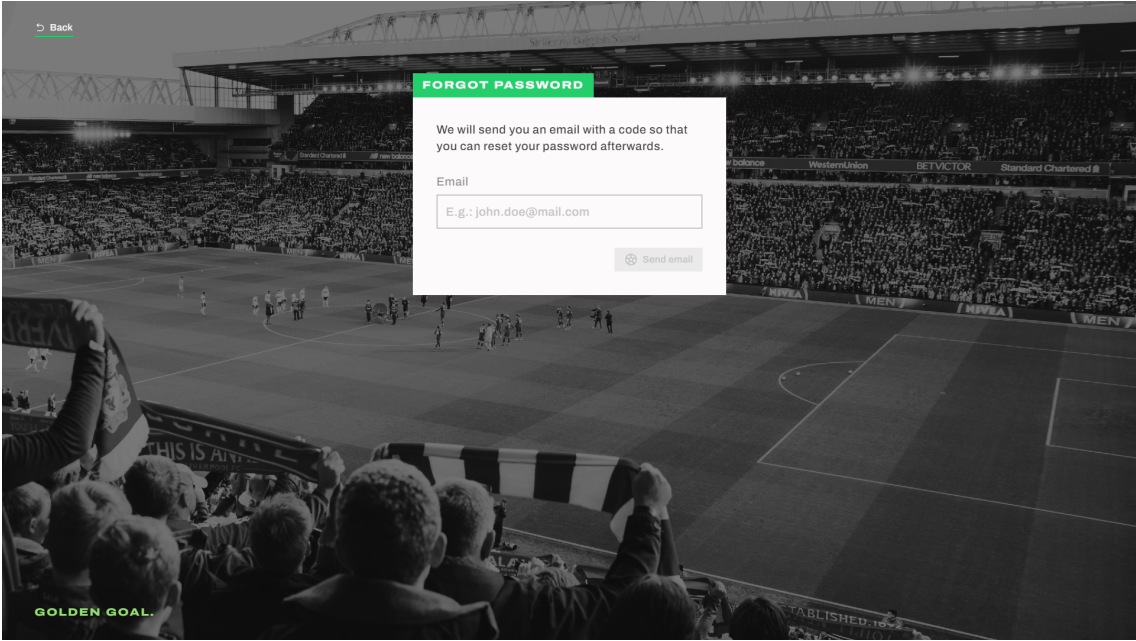


Figure C.3: Password recovery page

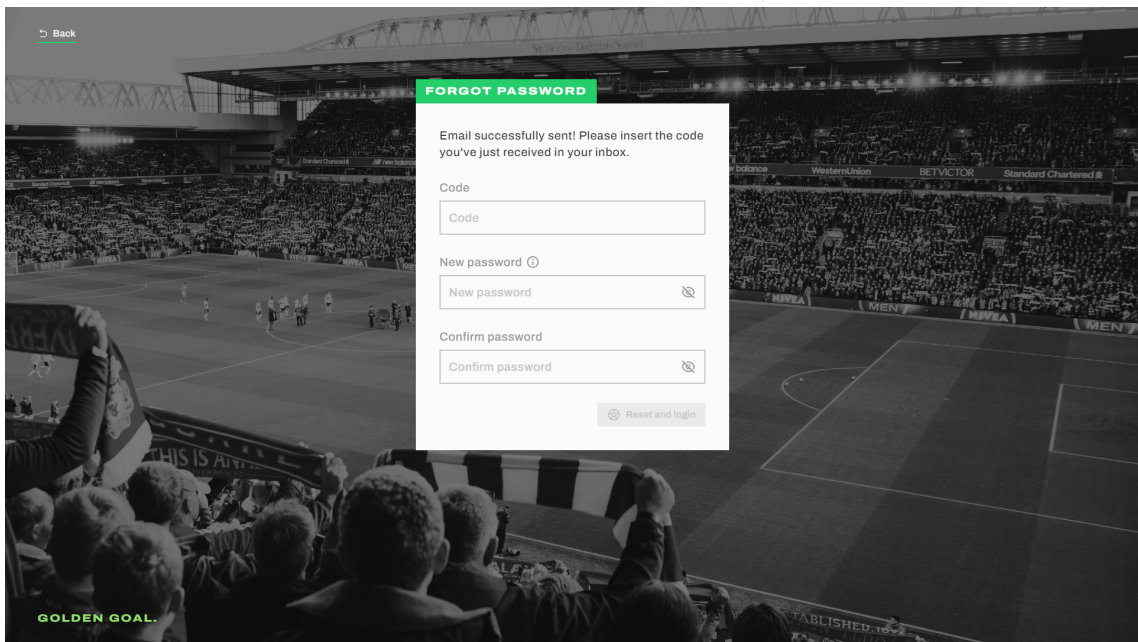


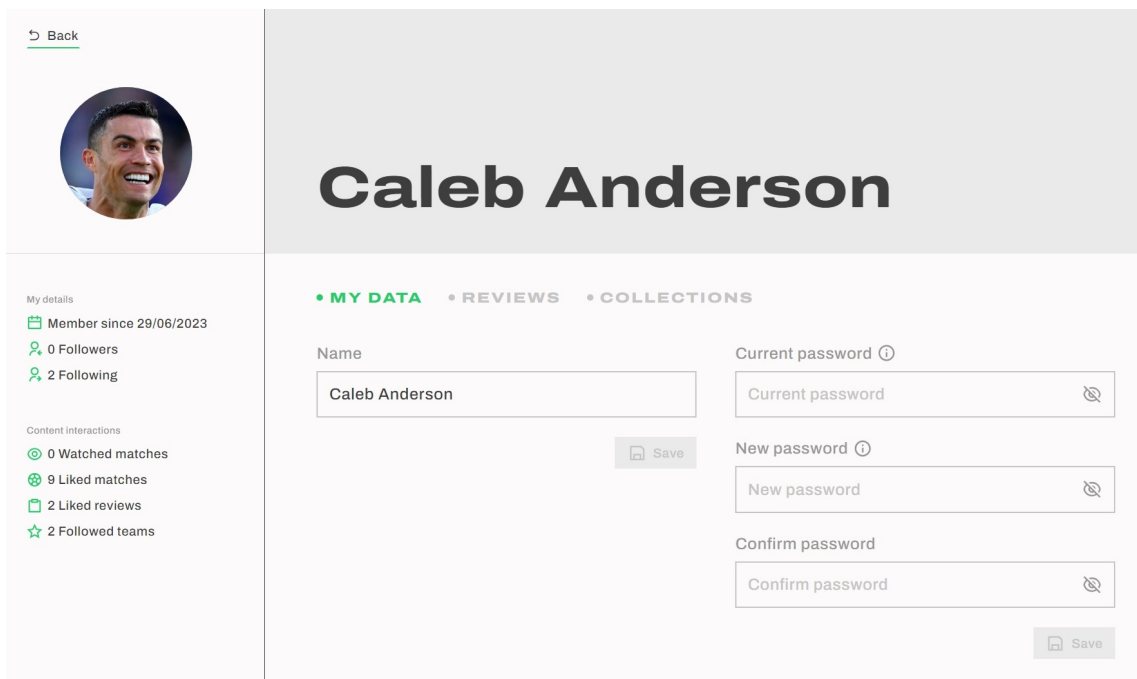
Figure C.4: Password recovery page 2

Public Profile Module

Figure C.5: Public Profile - Reviews tab

Figure C.6: Public Profile - Collections tab

Profile Module



Profile Module: Private Profile - My Data tab

Back

Caleb Anderson

• MY DATA • REVIEWS • COLLECTIONS

My details

- Member since 29/06/2023
- 0 Followers
- 2 Following

Content interactions

- 0 Watched matches
- 9 Liked matches
- 2 Liked reviews
- 2 Followed teams

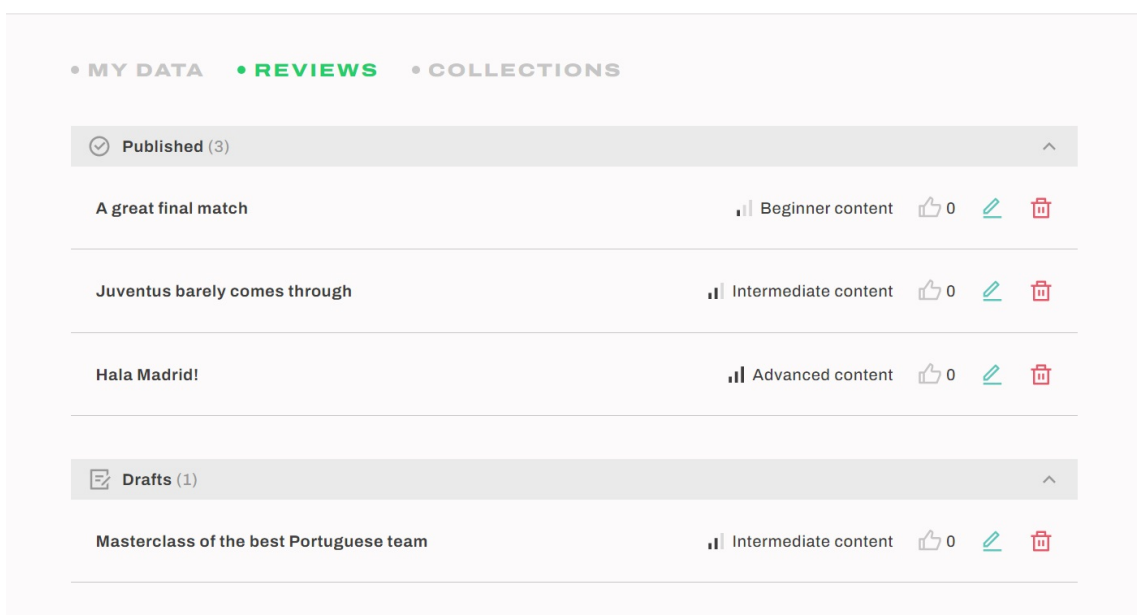
Name: Caleb Anderson [Save]

Current password [Save]

New password [Save]

Confirm password [Save]

Figure C.7: Private Profile - My Data tab



• MY DATA • REVIEWS • COLLECTIONS

Published (3)

- A great final match | Beginner content | 0 likes | [edit] [delete]
- Juventus barely comes through | Intermediate content | 0 likes | [edit] [delete]
- Hala Madrid! | Advanced content | 0 likes | [edit] [delete]

Drafts (1)

- Masterclass of the best Portuguese team | Intermediate content | 0 likes | [edit] [delete]

Figure C.8: Private Profile - Reviews tab

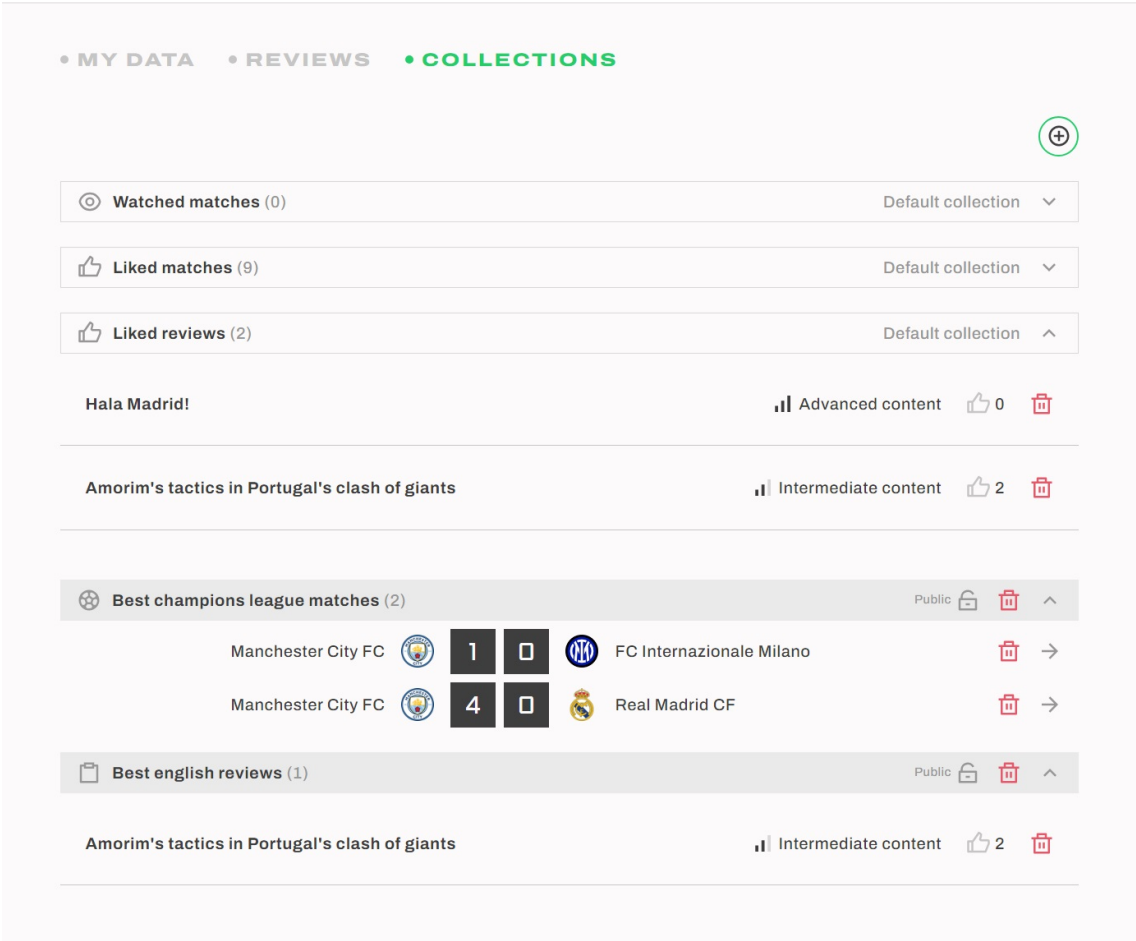


Figure C.9: Private Profile - Collections tab

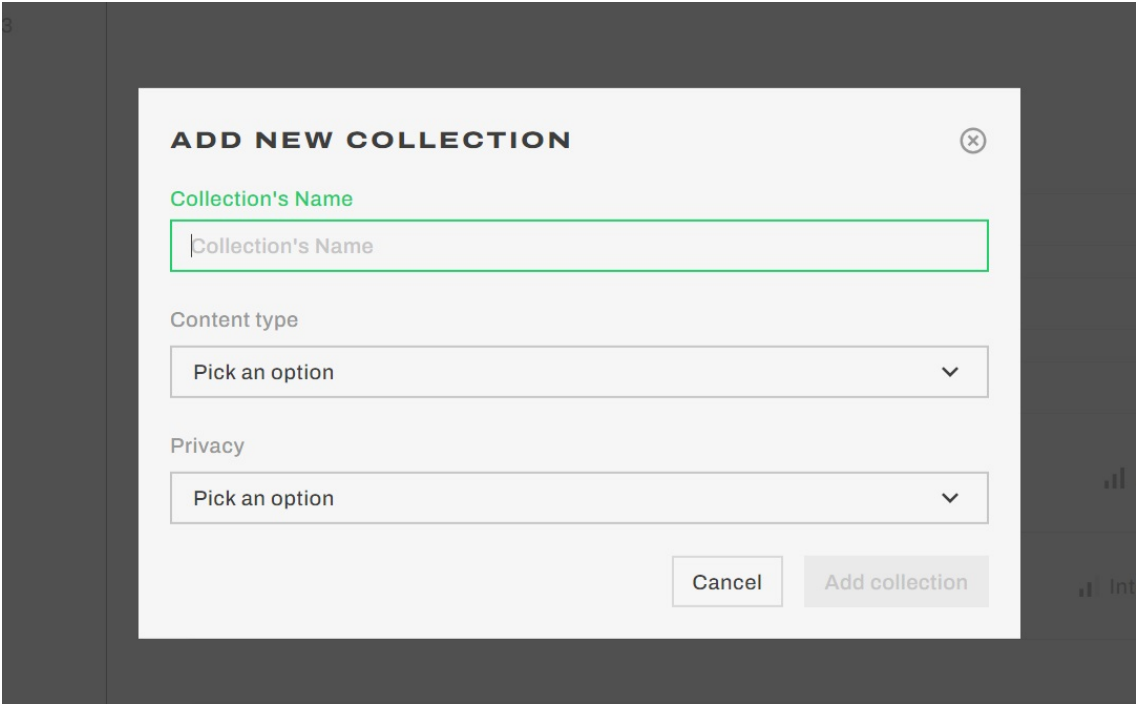


Figure C.10: Private Profile - Create Collection Modal

Reviews Module

[Back](#)

MOST POPULAR

- Amorim's tactics in Portugal's clash of giants**
Kiko Ramirez | 29/06/2023
Tags: Amorim, Benfica, +2
Score: 2 - 2
- Juventus barely comes through**
Caleb Anderson | 29/06/2023
Tags: Juve
Score: 0 - 1
- Hala Madrid!**
Caleb Anderson | 29/06/2023
Tags: Ronaldo Review
Score: 1 - 0

All reviews

Search reviews and tags

Filters

- Benfica and their fluidity**
Xavier Cooper | 30/08/2023
No tags found for this review...
Score: 2 - 2
- Why is the sporting formation so effective?**
Xavier Cooper | 30/08/2023
No tags found for this review...
Score: 2 - 2
- What a game! - Tactical analysis**
Nora Fuschell | 30/08/2023
No tags found for this review...
Score: 2 - 2
- One of my favorite games ever!**
Nora Fuschell | 30/08/2023
No tags found for this review...
Score: 2 - 2
- Analyzing the derby**
Harper Sullivan | 30/08/2023
No tags found for this review...
Score: 2 - 2
- Masterclass of the best Portuguese team**
Caleb Anderson | 30/08/2023
No tags found for this review...
Score: 1 - 2
- David vs Goliath**
Harper Sullivan | 29/06/2023
No tags found for this review...
Score: 1 - 0
- An immovable object vs an unstoppable force**
Xavier Cooper | 29/06/2023
No tags found for this review...
Score: 1 - 0
- Pep and the 4 center-backs**
Xavier Cooper | 29/06/2023
No tags found for this review...
Score: 1 - 0


1 2 3

POPULAR AUTHORS

- Kiko Ramirez**
3 followers
- Xavier Cooper**
1 follower
- Harper Sullivan**
1 follower
- Caleb Anderson**
0 followers
- Isabella Morgan**
0 followers
- Nora Fuschell**
0 followers

Figure C.11: All Reviews page


[Back](#)



Primeira Liga 21/05/2023
Estádio José Alvalade


Review details

Intermediate content 2

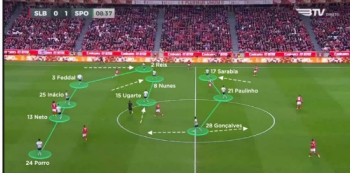


Kiko Ramirez
Member since 29/06/2023
3 followers

[Unfollow](#)



Rúben Amorim's tactical approach against Benfica showcased his ability to adapt and strategize effectively against strong opponents. Sporting CP, under Amorim's guidance, displayed a compact and disciplined defensive structure that proved difficult for Benfica to break down. The team defended as a cohesive unit, with players maintaining their positions and applying intense pressure on the ball carrier. This defensive solidity limited Benfica's attacking opportunities and frustrated their attempts to create meaningful chances.




EBL · Mar 8, 2023
@EBL2017 · [Follow](#)

Rúben Amorim of Sporting is regarded by many as the next big thing in coaching. I can understand why to an extent as he is clearly a talented coach.

However, Arteta's Arsenal are another level. Sporting could win tomorrow, but over the two legs, that's HIGHLY unlikely.

THREAD!



EBL · [Follow](#)

I was not familiar with Rúben Amorim before this thread, but I must admit that I am overwhelmed with his tactical qualities relative to the reputation he has...


Sporting play a 3-4-3 system similar to that of Conte at Spurs except with a supposedly more 'proactive' philosophy.

4:24 PM · Mar 8, 2023

328 · [Reply](#) · [Share](#)

[Read 2 replies](#)

Amorim's tactical flexibility was also evident in his in-game adjustments. As the match progressed, he made shrewd substitutions and alterations to the team's approach, ensuring that Sporting CP maintained control and minimized Benfica's impact. This adaptability showcased Amorim's tactical astuteness and his ability to read the game effectively.



Overall, Rúben Amorim's tactics against Benfica highlighted his strategic prowess and ability to create a well-organized and disciplined team. Sporting CP's defensive solidity, effective counter-attacking, and tactical flexibility were instrumental in their success against a strong opponent. Amorim's innovative methods and adaptability continue to make him one of the most promising young coaches in football today.

Written by [Kiko Ramirez](#) on 29/06/2023

Figure C.12: Review - Title, content, and details

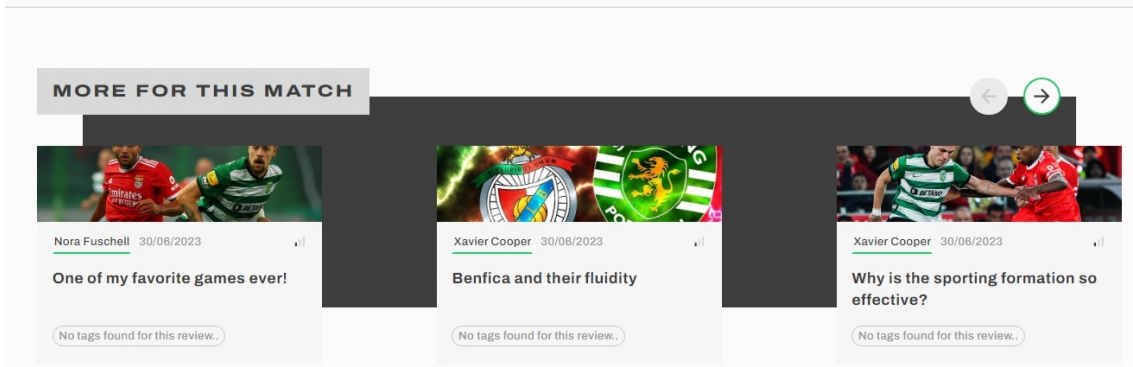
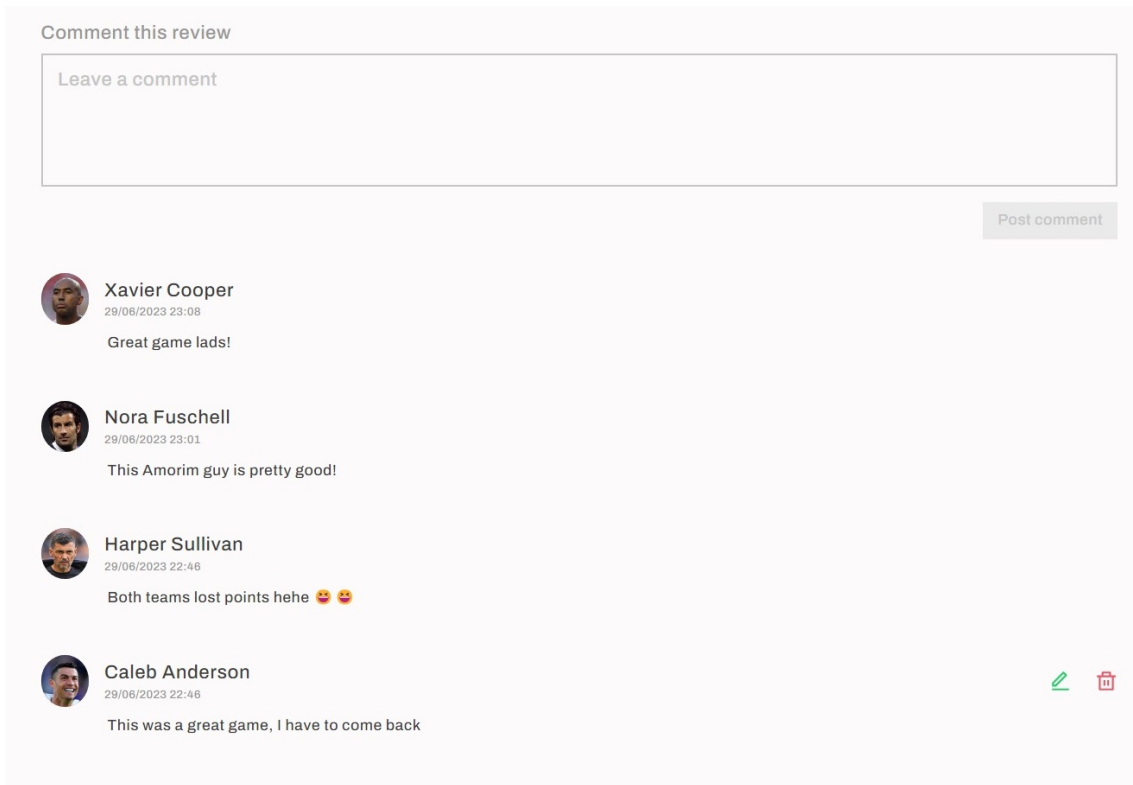


Figure C.13: Review - Comments and suggestions

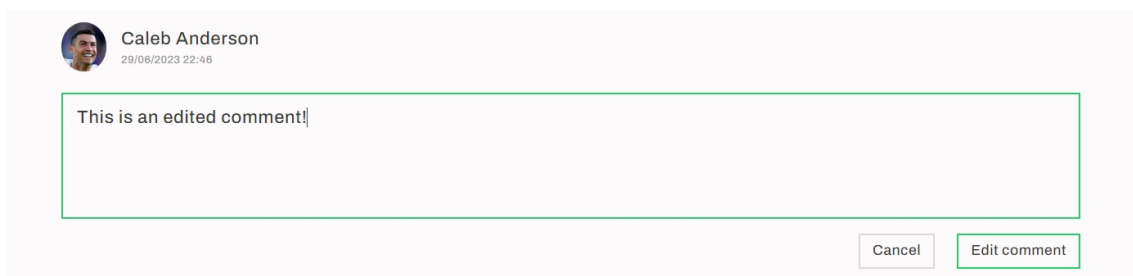


Figure C.14: Review - Edit comment

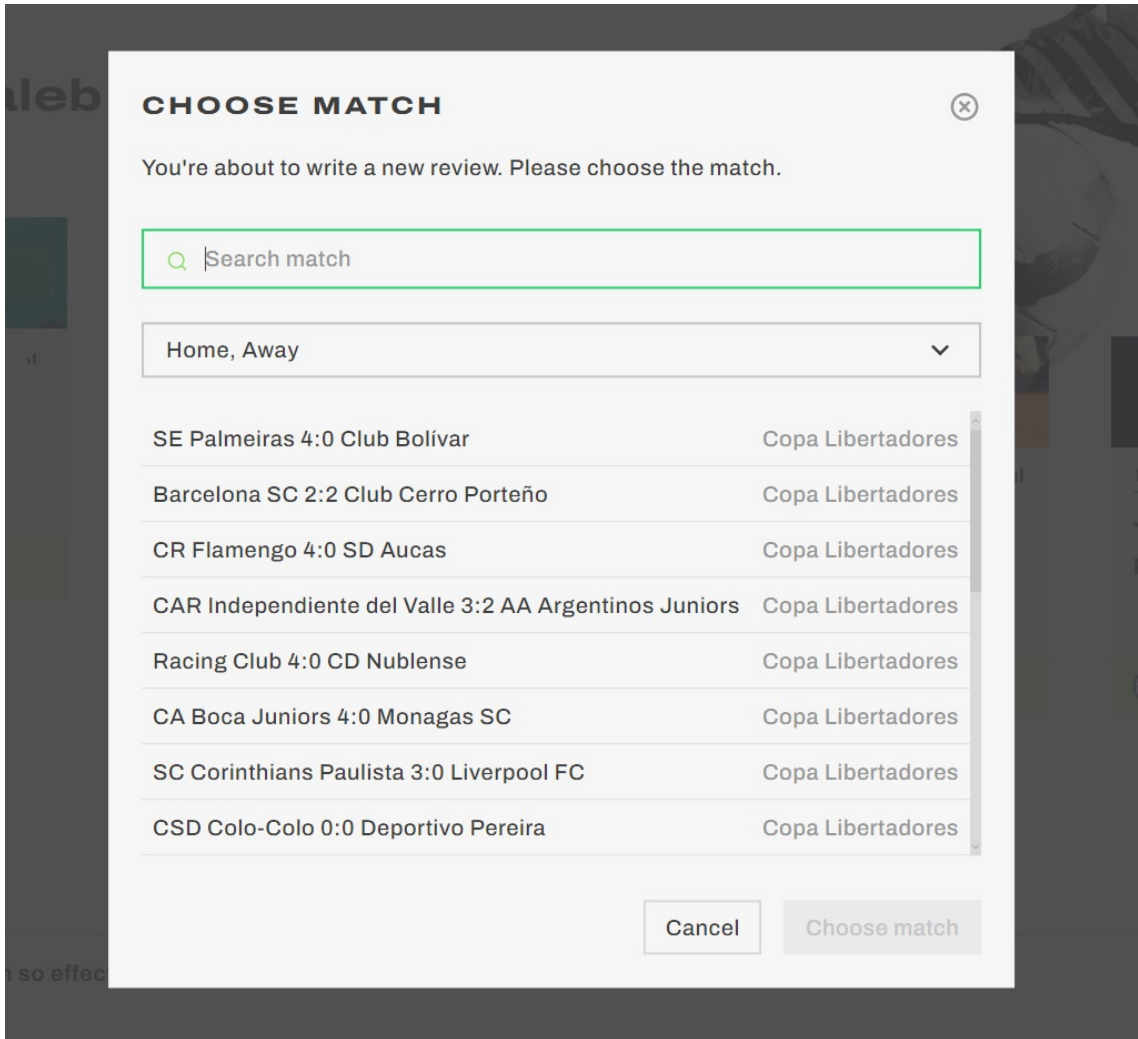


Figure C.15: Create review - Choose Match Modal

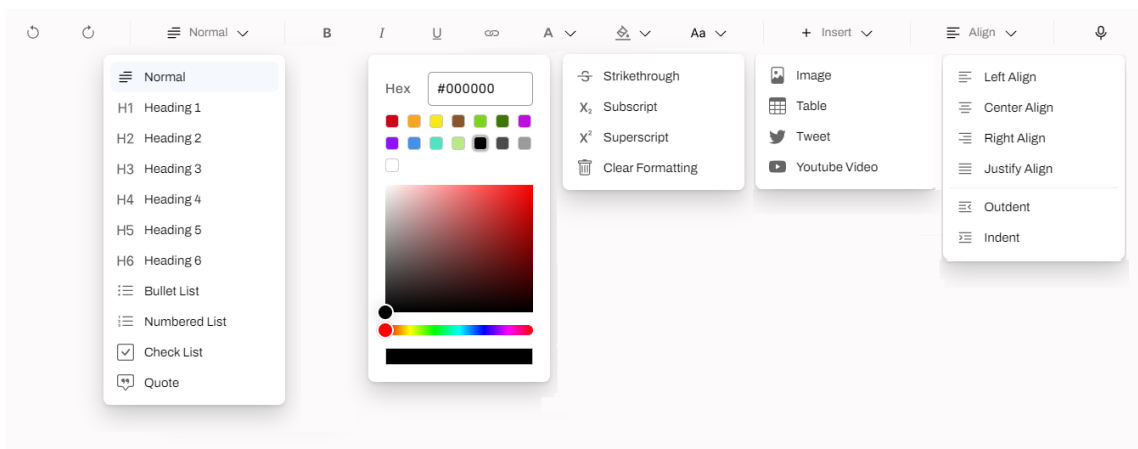


Figure C.16: Create review - Expanded editor toolbar

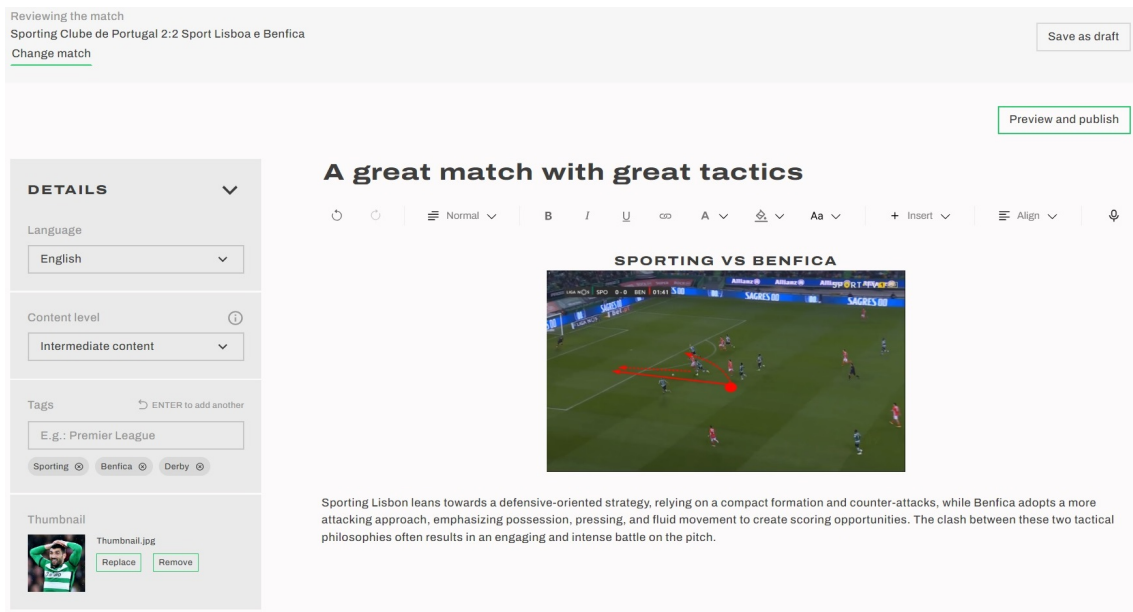


Figure C.17: Create review - Review editor



Figure C.18: Create review - Preview review

Matches Module

Back

All matches

Saturday 27 May 2023

Primeira Liga (7)

Gil Vicente FC	1	0	Casa Pia AC	Estádio Cidade de Barcelos
GD Chaves	1	4	Boavista FC	Estádio Municipal Eng. Manuel Branco Teixeira
Sport Lisboa e Benfica	3	0	CD Santa Clara	Estádio do Sport Lisboa e Benfica
Portimonense SC	0	2	FC Arouca	Estádio do Portimonense SC
GD Estoril Praia	3	1	CS Marítimo	Estádio António Coimbra da Mota
Sporting Clube de Braga	3	0	FC Paços de Ferreira	Estádio Municipal de Braga
FC Porto	3	0	Vitória SC	Estádio Do Dragão

Filters

Primeira Liga

Clear all

Friday 26 May 2023

Primeira Liga (2)

Bia Ave FC	2	2	FC Famalicão	Estádio de B. Ave Futebol
------------	---	---	--------------	---------------------------

Figure C.19: All Matches page

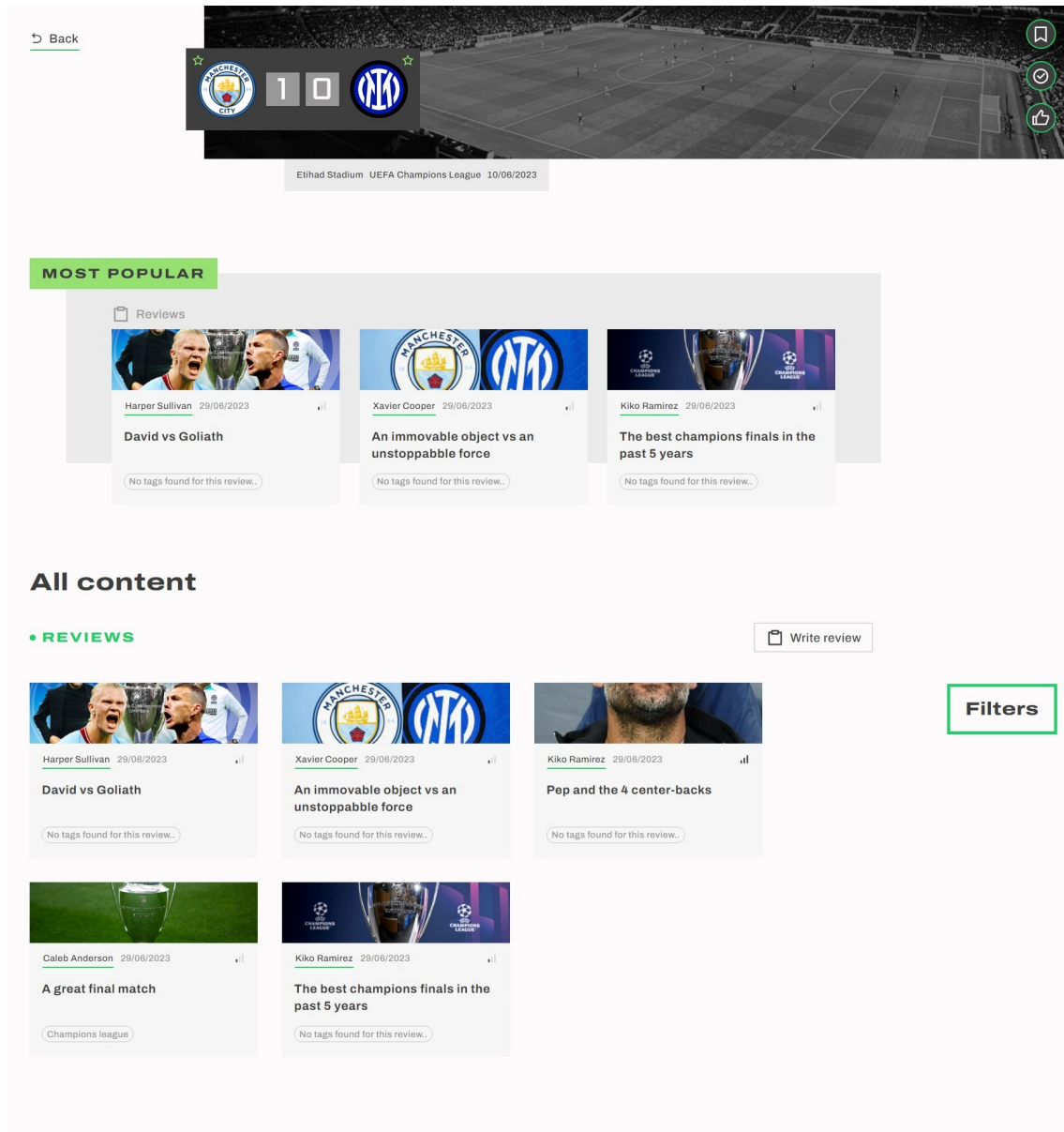


Figure C.20: Match page

Teams Module

Club details

- 2 Followers
- Portugal, 2020
- Estádio do Sport Lisboa e Benfica

• MATCHES • REVIEWS

Primeira Liga (34)

UEFA Champions League (14)

FC Internazionale Milano	3	3	Sport Lisboa e Benfica	19/04/2023
Sport Lisboa e Benfica	0	2	FC Internazionale Milano	11/04/2023
Sport Lisboa e Benfica	5	1	Club Brugge KV	07/03/2023
Club Brugge KV	0	2	Sport Lisboa e Benfica	15/02/2023
Maccabi Haifa FC	1	6	Sport Lisboa e Benfica	02/11/2022
Sport Lisboa e Benfica	4	3	Juventus FC	25/10/2022
Paris Saint-Germain FC	1	1	Sport Lisboa e Benfica	11/10/2022

Figure C.21: Club page - Matches tab

Club details

- 2 Followers
- Portugal, 2020
- Estádio do Sport Lisboa e Benfica

• MATCHES • **REVIEWS**

Amorim's tactics in Portugal's clash of giants
Kiko Ramirez 29/08/2023 Intermediate content 2

BENFICA GANHA LIGA PORTUGAL 22/23!
Nora Fuschell 29/08/2023 Advanced content 0

One of my favorite games ever!
Nora Fuschell 30/08/2023 Beginner content 0

Benfica and their fluidity
Xavier Cooper 30/08/2023 Beginner content 0

Why is the sporting formation so effective?
Xavier Cooper 30/08/2023 Beginner content 0

Figure C.22: Club page - Reviews tab

Homepage Module

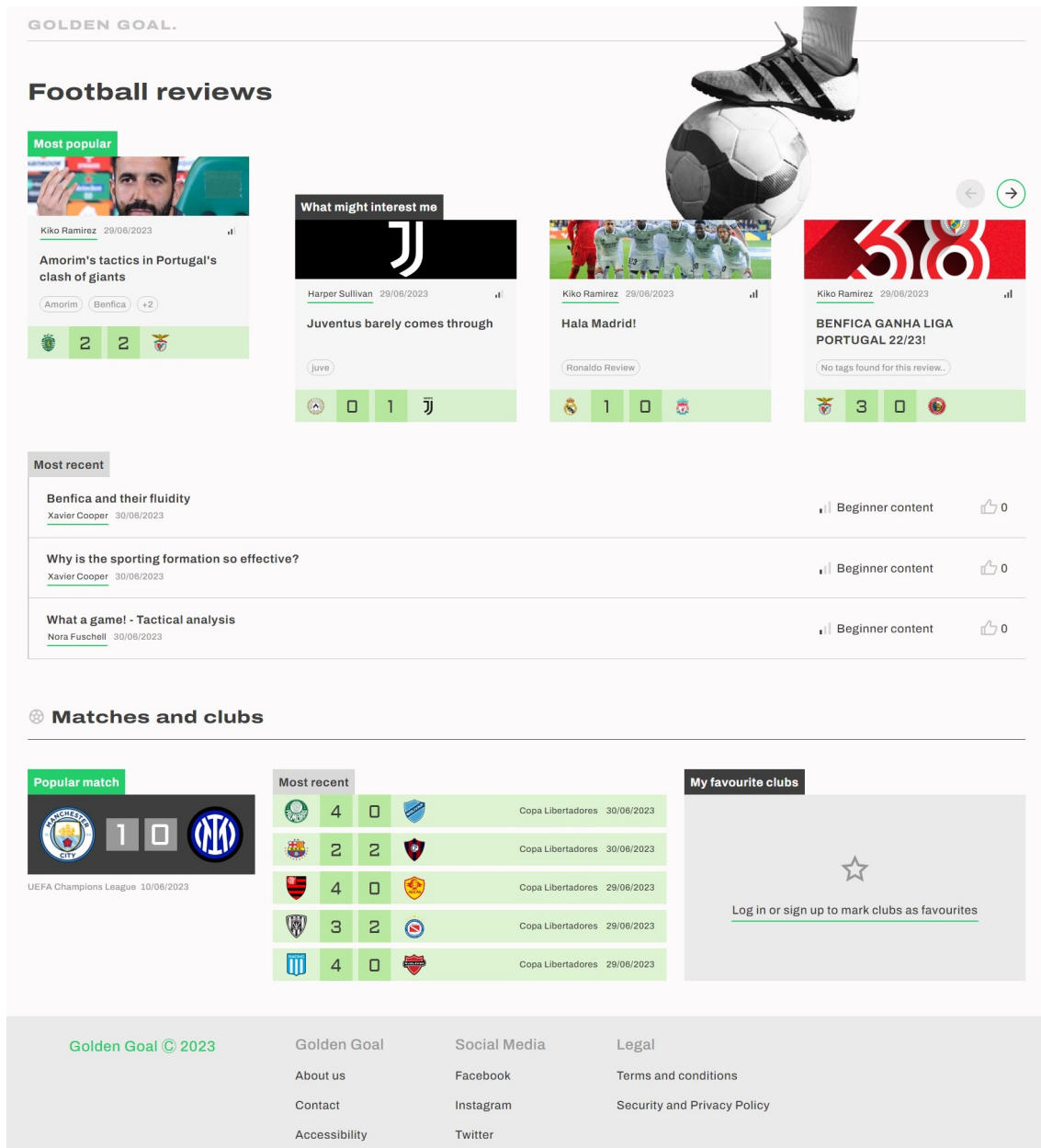



Figure C.23: Homepage for unauthenticated users

GOLDEN GOAL.

Welcome, Caleb Anderson

Most popular




Kiko Ramirez 29/06/2023

Amorim's tactics in Portugal's clash of giants

Amorim Benfica +2

2 2

From followed authors




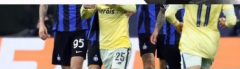
Harper Sullivan 29/06/2023

David vs Goliath

No tags found for this review...

1 0






Kiko Ramirez 29/06/2023

An undecided Porto falls to Inter

No tags found for this review...

0 0



Kiko Ramirez 29/06/2023

The best champions finals in the past 5 years

No tags found for this review...

1 0

Most recent

Benfica and their fluidity

Xavier Cooper 30/06/2023

Beginner content 0

Why is the sporting formation so effective?

Xavier Cooper 30/06/2023

Beginner content 0

Why is the sporting formation so effective?

Xavier Cooper 30/06/2023

Beginner content 0


What a game! - Tactical analysis

Nora Fuschelli 30/06/2023

Beginner content 0

Matches and clubs

Popular match



1 0

UEFA Champions League 10/06/2023

Most recent

4	0	Copa Libertadores	30/06/2023
2	2	Copa Libertadores	30/06/2023
4	0	Copa Libertadores	29/06/2023
3	2	Copa Libertadores	29/06/2023
4	0	Copa Libertadores	29/06/2023

My favourite clubs

1	2	Primeira Liga	26/05/2023
2	2	Primeira Liga	21/05/2023
4	0	UEFA Champions League	17/05/2023
2	1	Primeira Liga	13/05/2023
1	1	UEFA Champions League	09/05/2023

Golden Goal © 2023

About us

Contact

Accessibility

Golden Goal

Facebook

Instagram

Twitter

Social Media

Facebook

Instagram

Twitter

Legal

Terms and conditions

Security and Privacy Policy

Figure C.24: Homepage for authenticated users

Appendix D

Integration Tests

Authentication Module

ID	US	Scenario	Description	Steps	Expected Result	Result	Status
1	1	Register on the platform	An unregistered user wants to register himself on the platform	<ul style="list-style-type: none">- Go to registration page- Insert necessary information- Verify account in email	Account is verified successfully and the user can now log in	Account is verified successfully and the user can now log in	P
2	2	Log in to the platform	A unauthenticated user wants to login	<ul style="list-style-type: none">- Go to login page- Insert login information	User is successfully logged into the platform	User is successfully logged into the platform	P
3	2	Log in to the platform	Save login information	<ul style="list-style-type: none">- Go to login page- Insert login information- Select remember me checkbox	User information is saved and user is automatically logged in next time it opens the website	User information is saved and user is automatically logged in next time it opens the website	P
4	3	Recover password	A registered user wants to reset his password	<ul style="list-style-type: none">- Go to login page- Press forgot password button- Insert email associated with account- Receive recovery email- Define a new password	The user can log in with the his new password	The user can log in with his new password	P
5	4	Log out	An authenticated user wants to log out of the platform	<ul style="list-style-type: none">- Press the log out button	The user is logged out of the platform	The user is logged out of the platform	P

Table D.1: Authentication module integration test results

Personal Profile Module

ID	US	Scenario	Description	Steps	Expected Result	Result	Status
6	5	See personal information	A authenticated user wants to see his personal information	- Go to profile	User is displayed his personal information	User is displayed his personal information	P
7	6	Change name	A authenticated user wants to change his name	- Go to profile - Write new name - Submit form	User's name is updated	User's name is updated	P
8	7	Change position	A authenticated user wants to change his position	Not implemented			N/I
9	8	Change username	A authenticated user wants to change his username	Not implemented			N/I
10	9	Change email address	A authenticated user wants to change his email	Not implemented			N/I
11	10	Change password	A authenticated user wants to change his password	- Go to profile - Fill in the necessary fields - Submit form	User's password is updated	User's password is updated	P
12	11	Change profile picture	A authenticated user wants to change his profile picture	- Go to profile - Click the profile picture - Upload new profile picture	User's profile picture is updated	User's profile picture is updated	P
13	12	List collections	A authenticated user wants to list his own collections	- Go to profile - Open collections tab	User's collections are listed	User's collections are listed	P
14	13	Manage collections	A authenticated user wants to create a new collection	- Go to profile - Open collections tab - Press add collection button - Fill in the necessary fields	A new collection is created	A new collection is created	P
15	13	Manage collections	A authenticated user wants to delete a collection	- Go to profile - Open collections tab - Press delete button	The collection is deleted	The collection is deleted	P
16	13	Manage collections	A authenticated user wants to rename a collection	Not implemented			N/I
17	13	Manage collections	A authenticated user wants to change a collection's visibility	- Go to profile - Open collections tab - Press collection visibility button	The collection's visibility is changed	The collection's visibility is changed	P
18	14	Remove items	A authenticated user wants to remove items from a collection	- Go to profile - Open collections tab - Press item delete button	The item is removed from the collection	The item is removed from the collection	P

Table D.2: Personal Profile module integration test results - Part 1

ID	US	Scenario	Description	Steps	Expected Result	Result	Status
19	15	List reviews and drafts	A authenticated user wants to list his own reviews and drafts	- Go to profile - Open reviews tab	Reviews and drafts are listed	Reviews and drafts are listed	P
20	16	Delete a review or draft	A authenticated user wants to delete his review or draft	- Go to profile - Open reviews tab - Click review or draft delete button	The review or draft is deleted	The review or draft is deleted	P
21	17	See review's statistics	A authenticated user wants to see his review's statistics	Not implemented			N/I
22	18	See experiences and drafts	A authenticated user wants to see his experiences and drafts	Not implemented			N/I
23	19	Delete experiences and drafts	A authenticated user wants to delete his experiences or drafts	Not implemented			N/I
24	20	See experiences statistics	A authenticated user wants to see statistics about his experiences	Not implemented			N/I
25	21	See personal statistics	A authenticated user wants to see his own statistics	- Go to profile	The user can see his own statistics	The user can see his own statistics	P

Table D.3: Personal Profile module integration test results - Part 2

Public Profile Module

ID	US	Scenario	Description	Steps	Expected Result	Result	Status
26	22	See the author's personal information	A user wants to see the author's personal information	- Go to the author's profile	The user can see the author's information	The user can see the author's information	P
27	23	See the author's statistics	A user wants to see the author's statistics	- Go to the author's profile	The user can see the author's statistics	The user can see the author's statistics	P
28	24	Follow the author	A user wants to follow the author	- Go to the author's profile - Click the follow button	The author is added to the user's followed authors	The author is added to the user's followed authors	P
29	25	See the author's reviews	A user wants to see the author's reviews	- Go to the author's profile	The user can see the author's reviews	The user can see the author's reviews	P
30	25	See the author's reviews	A user wants to filter the author's reviews		Not implemented		N/I
31	25	See the author's reviews	A user wants to sort the author's reviews		Not implemented		N/I
32	26	See the author's experiences	A user wants to see the author's experiences		Not implemented		N/I

Table D.4: Public Profile module integration test results

Reviews Module

ID	US	Scenario	Description	Steps	Expected Result	Result	Status
33	27	See all reviews	A user wants to see all reviews	- Go to reviews page	The user can see all the reviews on the platform	The user can see all the reviews on the platform	P
34	27	See all reviews	A user wants to filter the review list	- Go to reviews page	The user can see the filtered reviews	The user can see the filtered reviews	P
35	27	See all reviews	A user wants to search reviews using text	- Go to reviews page	The user can see the reviews filtered by the search text	The user can see the reviews filtered by the search text	P
36	27	See all reviews	A user wants to sort reviews		Not implemented		N/I
37	28	See trending reviews	A user wants to see the trending reviews	- Go to reviews page	The user can see trending reviews	The user can see trending reviews	P
38	29	See recent reviews by followed author	An authenticated user wants to see recent reviews by authors he follows	- Go to reviews page	The authenticated user can see recent reviews by the authors he follows	The authenticated user can see recent reviews by the authors he follows	P
39	30	See popular authors	A user wants to find the most popular authors	- Go to reviews page	The user can see the most popular authors	The user can see the most popular authors	P
40	31	Navigate through tags	A user wants to navigate through tags to find similar content		Not implemented		N/I
41	32	See a full review	A user wants to see the content of the review	- Open a review	The user can see the content of the review	The user can see the content of the review	P
42	32	See a full review	A user wants to see the details of the review	- Open a review	The user can see the details of the review	The user can see the details of the review	P
43	32	See a full review	A user wants to see the match associated with the review	- Open a review	The user can see the match associated with the review	The user can see the match associated with the review	P
44	33	Share on social media	A user wants to share a review on social media		Not implemented		N/I
45	34	Comment a review	An authenticated user wants to comment on a review	- Open a review - Write the comment - Post the comment	The comment is added to the review	The comment is added to the review	P
46	34	Comment a review	An authenticated user wants to edit one of his comments	- Open a review - Click on the edit button of the comment - Edit the comment text	The comment text is edited	The comment text is edited	P
47	34	Comment a review	An authenticated user wants to delete one of his comments	- Open a review - Click on the delete button of the comment	The comment is deleted from the review	The comment is deleted from the review	P

Table D.5: Reviews module integration test results - Part 1

ID	US	Scenario	Description	Steps	Expected Result	Result	Status
48	35	Like a review	An authenticated user wants to like a review	- Open a review - Press the like button	The review is liked	The review is liked	P
49	36	Follow the review's author	An authenticated user wants to follow the review's author	- Open a review - Press the follow button	The author of the review is followed	The author of the review is followed	P
50	37	Add the review to a collection	An authenticated user wants to add the review to one of his collections	- Open a review - Press the add to collection button - Select the collection to add the review	The review is added to the collection	The review is added to the collection	P
51	38	Create a review	A authenticated user wants to associate a match to his review	- Press add review button - Select a match from the match list	Review is associated with the selected match	Review is associated with the selected match	P
52	38	Create a review	A authenticated user wants to add content to his review	- Press add review button - Select a match from the match list - Add content to review	The content is added to the review	The content is added to the review	P
53	38	Create a review	A authenticated user wants to set his review's details	- Press add review button - Select a match from the match list - Set review details	The details are now set in the review	The details are now set in the review	P
54	38	Create a review	A authenticated user wants to preview review	- Press add review button - Select a match from the match list - Add content to review - Set review details - Press preview button	A preview of the review is displayed to the user	A preview of the review is displayed to the user	P
55	38	Create a review	A authenticated user wants to publish his review	- Press add review button - Select a match from the match list - Add content to review - Set review details - Press preview button - Press publish button	The review is published	The review is published	P
56	38	Create a review	A authenticated user wants to save his review as a draft	- Press add review button - Select a match from the match list - Press save as draft button	A new draft is created with the review's content	A new draft is created with the review's content	P

Table D.6: Reviews module integration test results - Part 2

Matches Module

ID	US	Scenario	Description	Steps	Expected Result	Result	Status
57	47	See all matches	A user wants to see all the matches present on the platform	- Go to matches page	The user sees all the matches available on the platform	The user can see all the matches available on the platform	P
58	47	See all matches	A user wants to see all matches grouped by date and competition	- Go to matches page	The user sees the matches grouped by page and competition	The user can see the matches grouped by page and competition	P
59	47	See all matches	A user wants to filter the match list	- Go to matches page - Open filters menu - Apply filters	The user sees the matches corresponding to his filters	The user sees the matches corresponding to his filters	P
60	47	See all matches	A user wants to search for a match using text	- Go to matches page - Insert text on the search bar	The user sees the matches corresponding to his search text	The user sees the matches corresponding to his search text	P
61	48	See match	A user wants to see information about a specific match	- Open match page	The user sees information about the match he opened	The user sees information about the match he opened	P
62	49	See match details	A user wants to see the details about a match	- Open match page	The user sees the details of the match he opened	The user sees the details of the match he opened	P
63	50	Mark match as watched	A user wants to mark a match as watched	- Open match page - Click on the "mark as watched" button	The match is now marked as watched	The match is now marked as watched	P
64	51	Add match to collection	A user wants to add a match to one of his collections	- Open match page - Click on the "add to collection" button - Select the collection to add the match to	The match is now present in the collection	The match is now present in the collection	P
65	52	Like match	A user wants to like a match	- Open match page - Click on the like button	The match is liked	The match is liked	P
66	53	Follow team	A user wants to follow a team present on a match	- Open match page - Click on the follow team button	The team is present on his followed teams list	The team is present on his followed teams list	P
67	54	See related reviews	A user wants to see reviews related to a match	- Open match page	The user can see reviews from the same match as the one he opened	The user can see reviews from the same match as the one he opened	P
68	54	See related reviews	A user wants to see the most popular reviews related to the match	- Open match page	The user can see the most popular reviews associated with the opened match	The user can see the most popular reviews associated with the opened match	P
69	54	See related reviews	A user wants to filter the reviews related to the match	- Open match page - Open filters menu - Apply filters	The user only sees the reviews corresponding to his filters	The user only sees the reviews corresponding to his filters	P
70	55	See related experiences	A user wants to see experiences related to the match		Not implemented		N/I

Table D.7: Matches module integration test results

Teams Module

ID	US	Scenario	Description	Steps	Expected Result	Result	Status
71	56	See team information	A user wants to see information about a team	- Open team page	The user sees the information about the team he just opened	The user sees the information about the team he just opened	P
72	57	See a team's matches	A user wants to see the matches of a team	- Open team page	The user sees the matches that the team has played	The user sees the matches that the team has played	P
73	57	See a team's matches	A user wants to see the team's matches grouped by competition	- Open team page	The user sees the matches grouped by competition	The user sees the matches grouped by competition	P
74	57	See a team's matches	A user wants to filter the team's matches by season	Not implemented			N/I
75	58	See a team's statistics	A user wants to see statistics about the team	Not implemented			N/I
76	59	See related reviews	A user wants to see the reviews related to the team	- Open team page - Open reviews tab	The user sees the reviews associated with the team	The user sees the reviews associated with the team	P
77	59	See related reviews	A user wants to filter the review list by the associated match's season	Not implemented			N/I
78	60	See a team's experiences	A user wants to see the experiences related to the team	Not implemented			N/I
79	61	Follow team	A authenticated user wants to follow the team	- Open team page - Press the follow team button	The team is added to the followed teams list	The team is added to the followed teams list	P

Table D.8: Teams module integration test results

Homepage Module

ID	US	Scenario	Description	Steps	Expected Result	Result	Status
80	62	See reviews	A user wants to see the most popular reviews	- Open homepage	The user can see the most popular reviews	The user can see the most popular reviews	P
81	62	See reviews	A user wants to see the most recent reviews	- Open homepage	The user can see the most recent reviews	The user can see the most recent reviews	P
82	62	See reviews	An authenticated user wants to see reviews from followed authors	- Open homepage	The authenticated user can see the reviews from the authors he follows	The authenticated user can see the reviews from the authors he follows	P
83	63	See experiences	A user wants to see experiences		Not implemented		N/I
84	64	See matches	A user wants to see the most popular matches	- Open homepage	The user can see the most popular matches	The user can see the most popular matches	P
85	64	See matches	A user wants to see the most recent matches	- Open homepage	The user can see the most recent matches	The user can see the most recent matches	P
86	64	See matches	An authenticated user wants to see matches from followed teams	- Open homepage	The authenticated user can see the matches from the teams he follows	The authenticated user can see the matches from the teams he follows	P
87	65	See teams	An authenticated user wants to see statistics about their favorite teams		Not implemented		N/I
88	66	Search menu	A user wants to search for content on the platform		Not implemented		N/I

Table D.9: Homepage module integration test results

