CoSpace
Software Design Description

Revision History

Version	*Date*	*Author(s)*	*Change Description*
0.1	13.05.2021	Mert Can Demir, M. Şamil Ateşoğlu	Initial version of the SDD document.

1 INTRODUCTION

1.1 Purpose and Scope

This document provides explanation and necessary yet detailed enough information about the project CoSpace. It also explains how a particular piece of the software developed. In doing so, various tools were used, such as diagrams, some design-related illustrations, and various other tools that show the structure of the software. Furthermore, according to our scope, this document aims to describe the project's design views that will meet the functional and non-functional requirements indicated in the software's Software Requirements Specification.

1.2 Document Overview

This document consists of the chapters following:

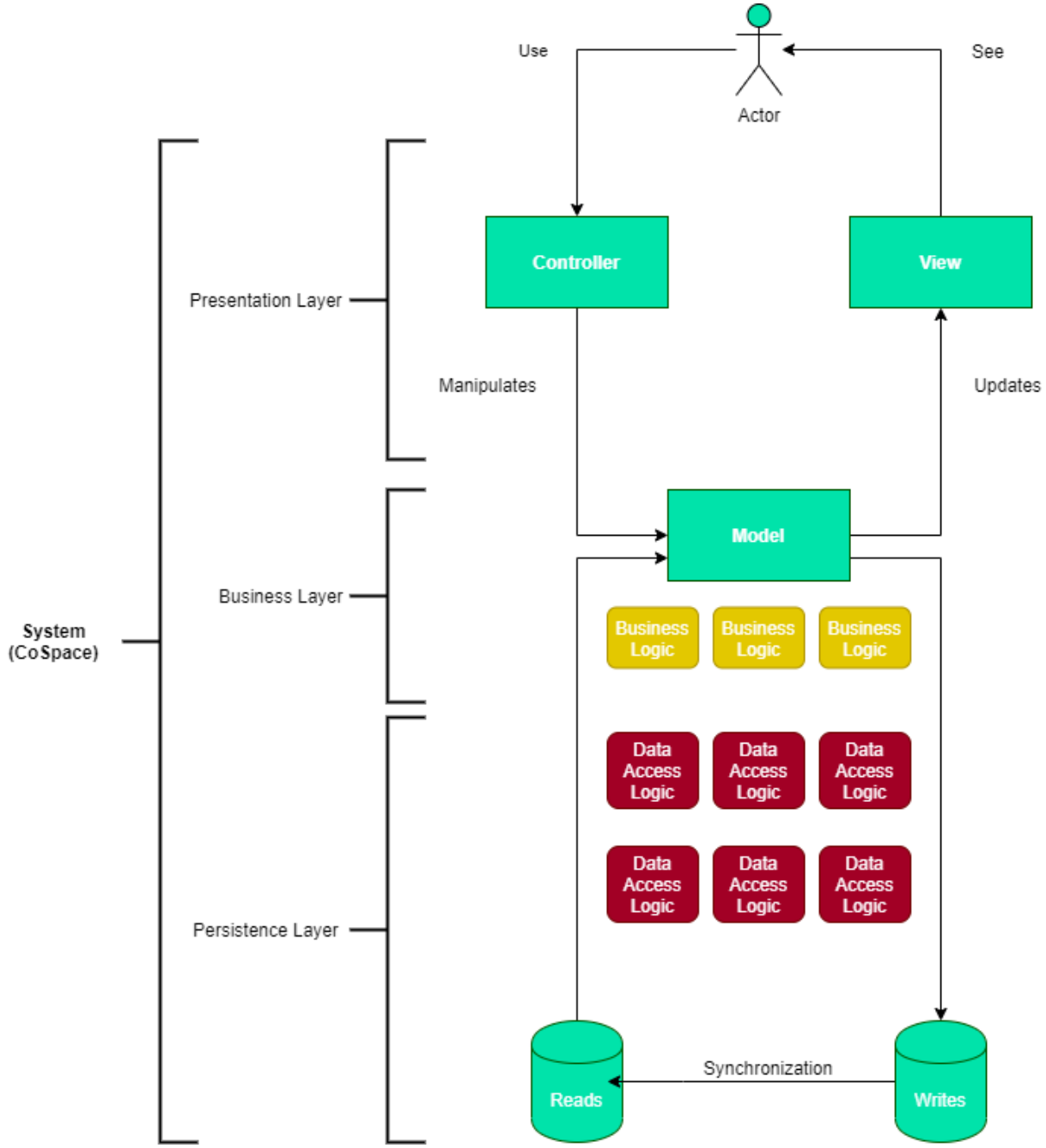
- In the Revision History, the changes over the document are recorded.
- In the Introduction part, which is part of this part, the general information about the document is described. This part consists of five sub-parts. Purpose and Scope gives a general intuition about the document. Document Overview is responsible for giving the information about the chapters of the document, a little more detailed than Introduction. System Overview gives a general description over the complete system preferably with some drawing. Definitions, Acronyms, and Abbreviations is responsible for containing the terminology specific for this document which are not standard. References show the references used.
- Design Constraints and Decisions part shows some constraints and given decisions with some detail.
- Design Details have the drawings, which are Class diagram, Sequence diagram for use-cases specified on System-Wide Requirements Specification Document, Entity-Relationship diagram, and interfaces described for the system.
- Requirements Traceability part ensures that all specifications are fulfilled by maintaining a traceability relationship between requirements and program design.
- Annexes is an extra part, in case there are some parts of the document that need to be explained in more detail.

1.3 System Overview

CoSpace is an online social interest e-club web application. The application provides a common space for users to engage in conversations about certain topics, form clubs and sub-clubs (a more specific club under a parent club), plan online or onsite events regarding their sub-clubs.

The application is built on open-source frameworks such as React, Spring and database PostgreSQL. Also, the source of the application is available on GitHub. Unfortunately, due to regulations, the source code of CoSpace is not available as public at the moment.

The system is created with two main components as frontend and backend. React is used for frontend, Spring is used for backend. The communication between frontend and backend is established with RESTful API.



In the system, MVC pattern is used. The system has Controller, Model and View separated. All of the requests created by the user are coming first to the controller, even though we have separate classes for controllers for cases i.e. authorization, club, post, etc. in the same package. Controller passes the related information to the View or Model components, gets the results from these, and responds to the user. In that way, we have an architecture that is not cluttered. Also, layered architecture is used for the application. View and Controller stays in the Presentation Layer. All of the other layers as Business Layer and Persistence Layer are in the Model. In Presentation Layer, the requests made by a user (it may be a login request, post request, create club request, logout request, etc.) are handled and converted for other layers (Model) if needed. In Business Layer, the business logic is handled and all of the authorization and validation is performed. In the Persistence Layer, all of the storage logic is stored and handled. Also, CRUD operations are performed there.

1.4 Definitions, Acronyms, and Abbreviations

Term/Acronym	*Definition*
SDD	Software Design Description

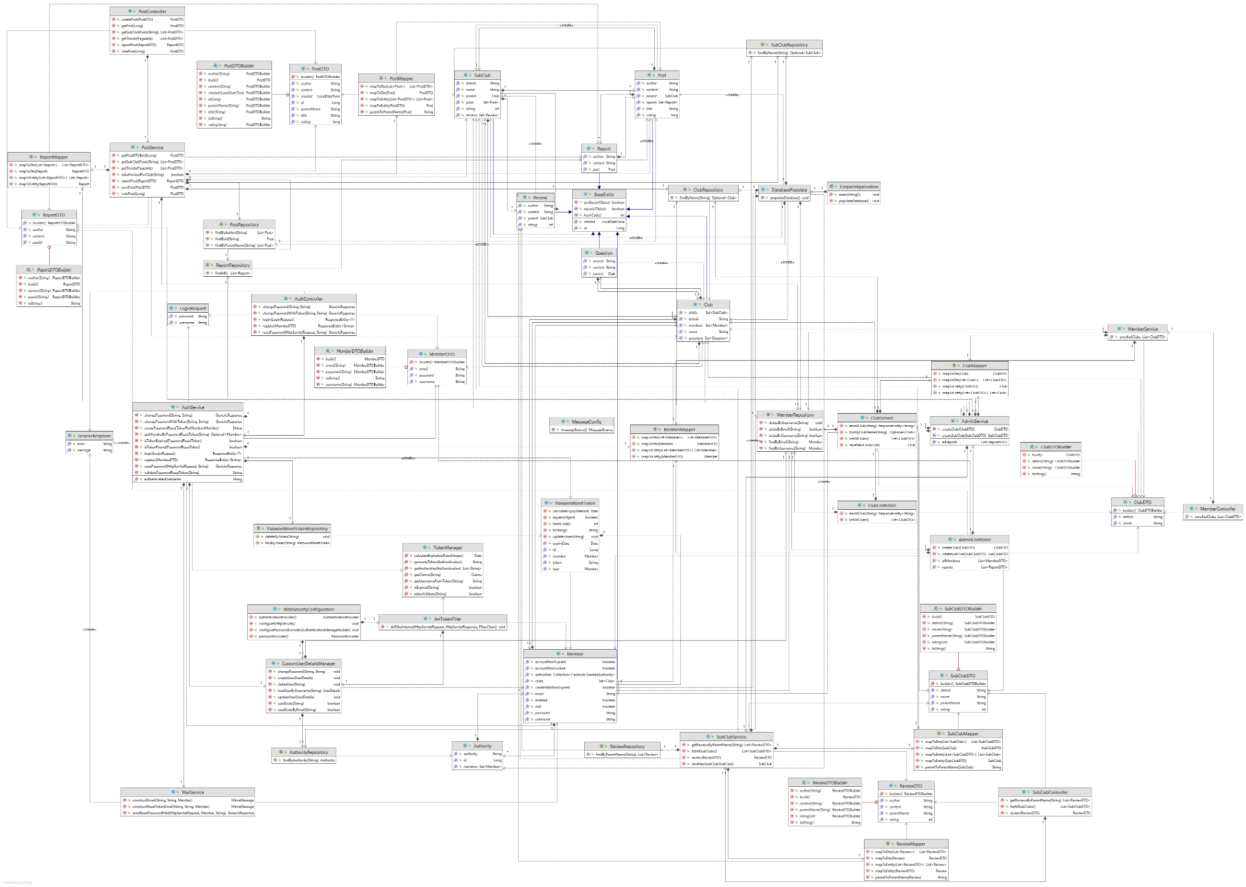
2 Design Constraints and Decisions

- React is used for the frontend, Spring is used for the backend of the application, as with these frameworks, applications can be released quickly, and thus time is saved. In addition, all of the design that belongs to GUI is created with React. Furthermore, the connection between frontend and backend is provided by a RESTful API with Spring Boot, as the application is a RESTful web service, and as this API provides CRUD operations as expected.
- PostgreSQL is used for the database, as PostgreSQL is an object-relational database. User-defined objects and behaviors such as data types, functions, operators, domains, and directories are an important feature of the object-relational database. From that perspective of view, PostgreSQL is a better choice among other relational databases, at least for our system.
- MVC pattern is used. We have Controller, Model and View separated. In that way, we have an architecture that is not cluttered.
- Layered architecture is implemented. We used Spring Framework for our backend. Thus, it is natural that we also used layered architecture, as the implementation of this architecture is easier, thus more time is saved.
- The application is created to be expected to work in any Chromium-based browsers or Firefox browser, as more than 90% of users use these browsers.
- The system is expected to provide 99% uptime for availability. The application should not be interrupted any longer. For uptime, we cannot choose any higher, because of higher cost.
- For any response created by the system, the response time should not exceed 400 ms, excluding the internet response time of the user. As the Doherty Threshold says, an immediate interaction between user and computer will heighten the experience and result in a much simpler and more enjoyable experience and it says that the user experience turns from painful to addictive after the system feedback time drops below 400ms. Hence, a response time should not exceed 400 milliseconds.
- To use the system, internet connection is required, as the application is a web application.

3 Design Details

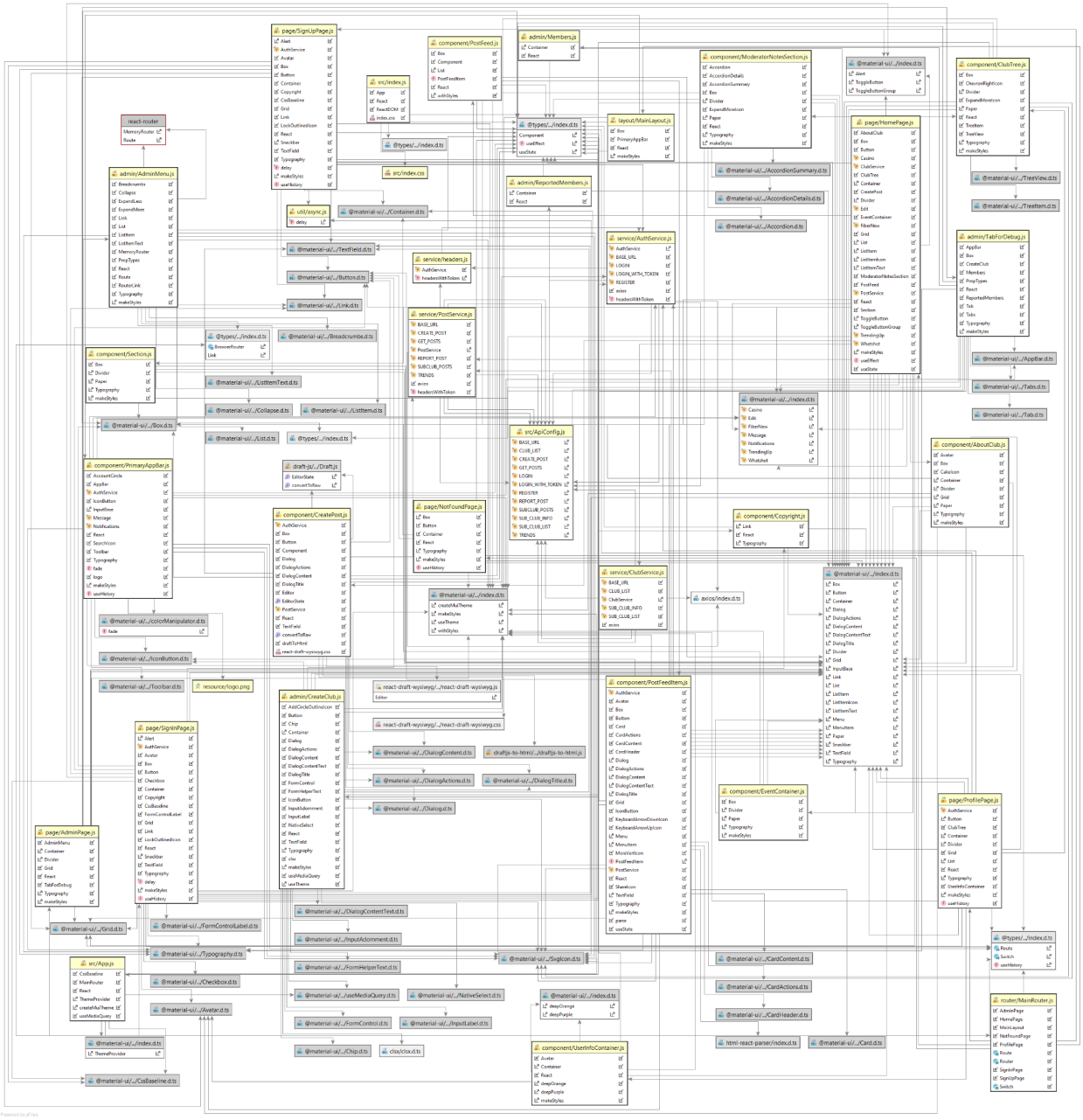
3.1 Software Components

3.1.1 CoSpace Back-End Service Class Diagram



Refer to here for high resolution version.

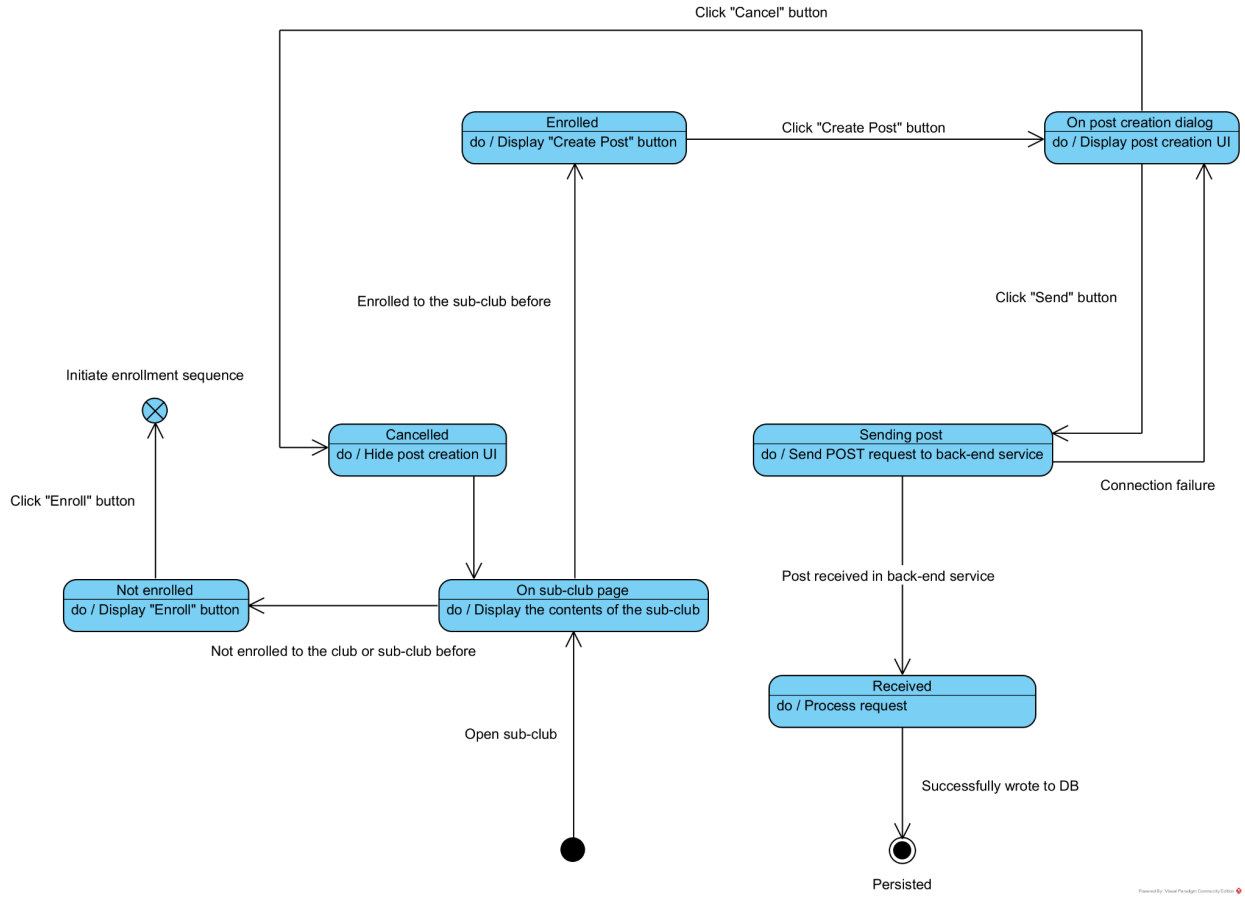
3.1.2 CoSpace Front-End Service Class Diagram



Refer to here for high resolution version.

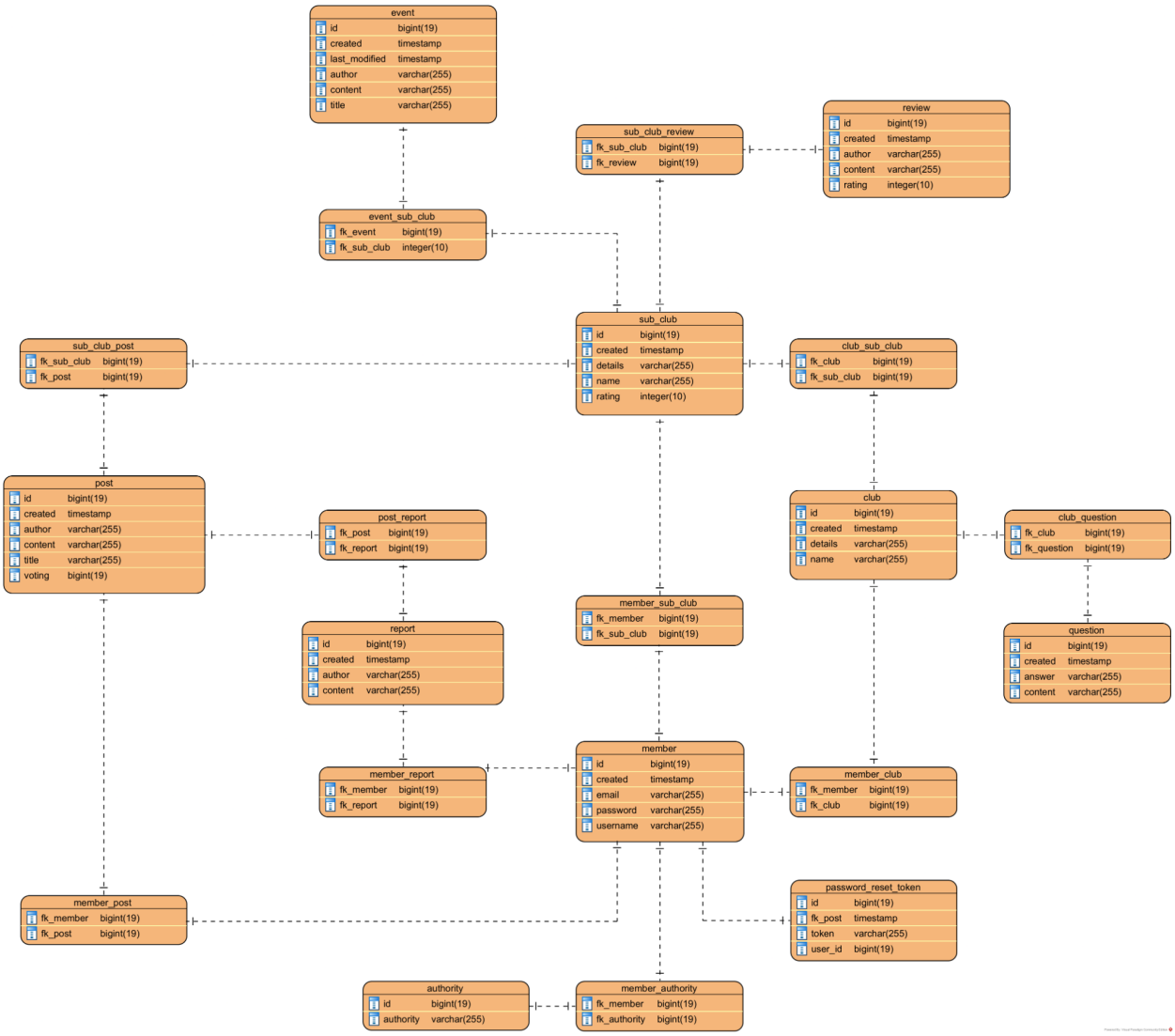
3.2 Software Behavior

3.2.1 State Diagram: UC-6 Create a Post



3.3 Data Model (E-R Diagram)

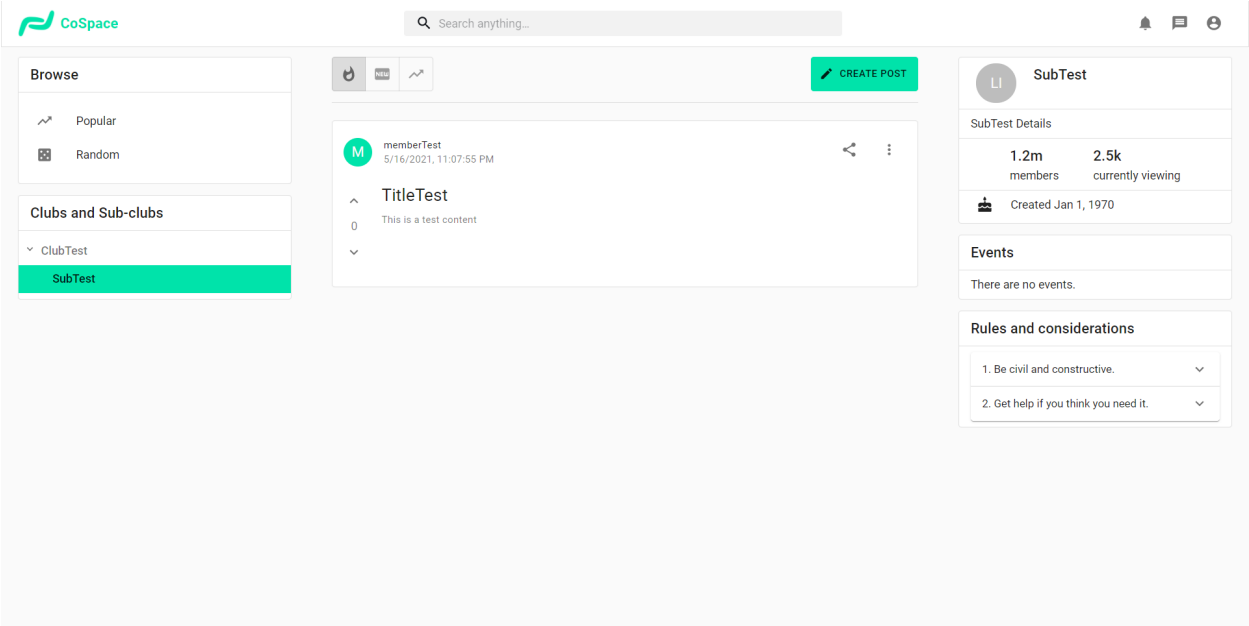
3.3.1 CoSpace Logical Entity Relationship Diagram



3.4 User Interface Design

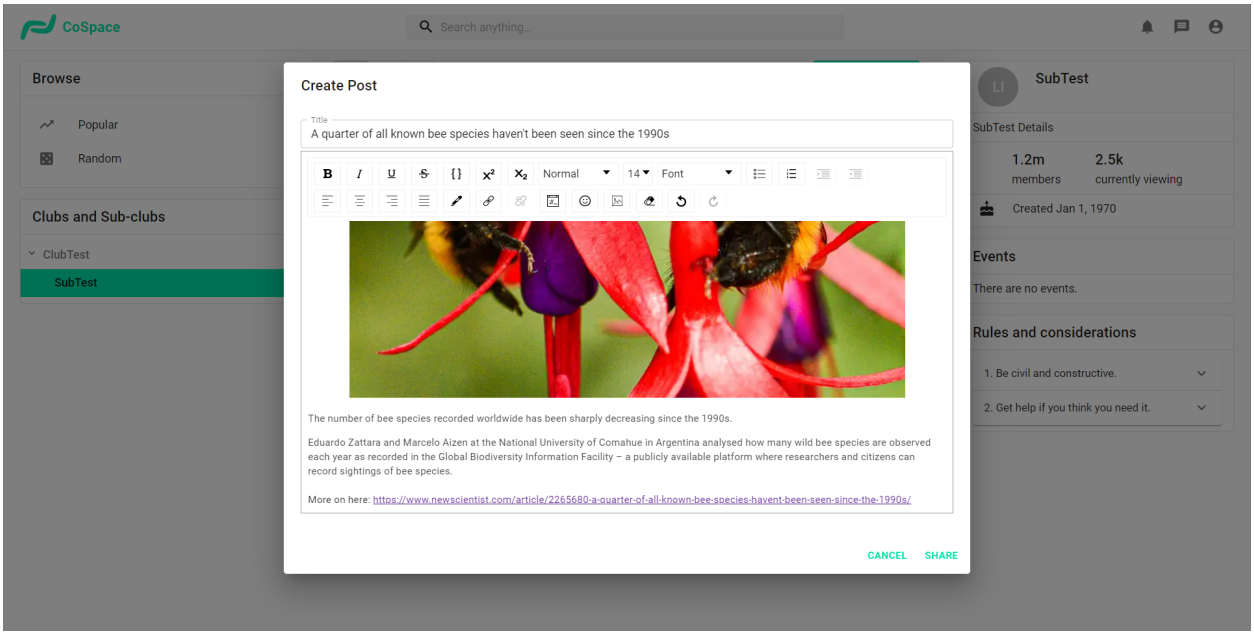
3.4.1 Main Page & Feed

The feed section, the club/sub-club tree, and their informations:

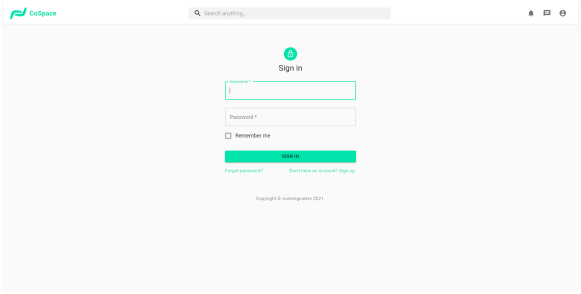


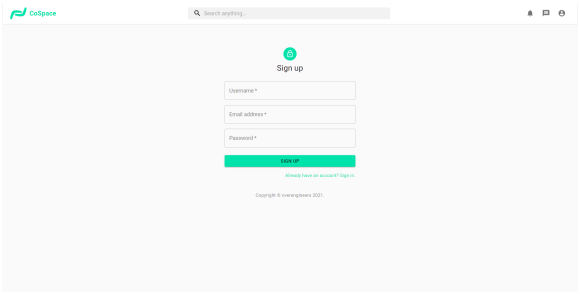
3.4.2 Creating a Post

The graphical user interface when creating a post (public message):

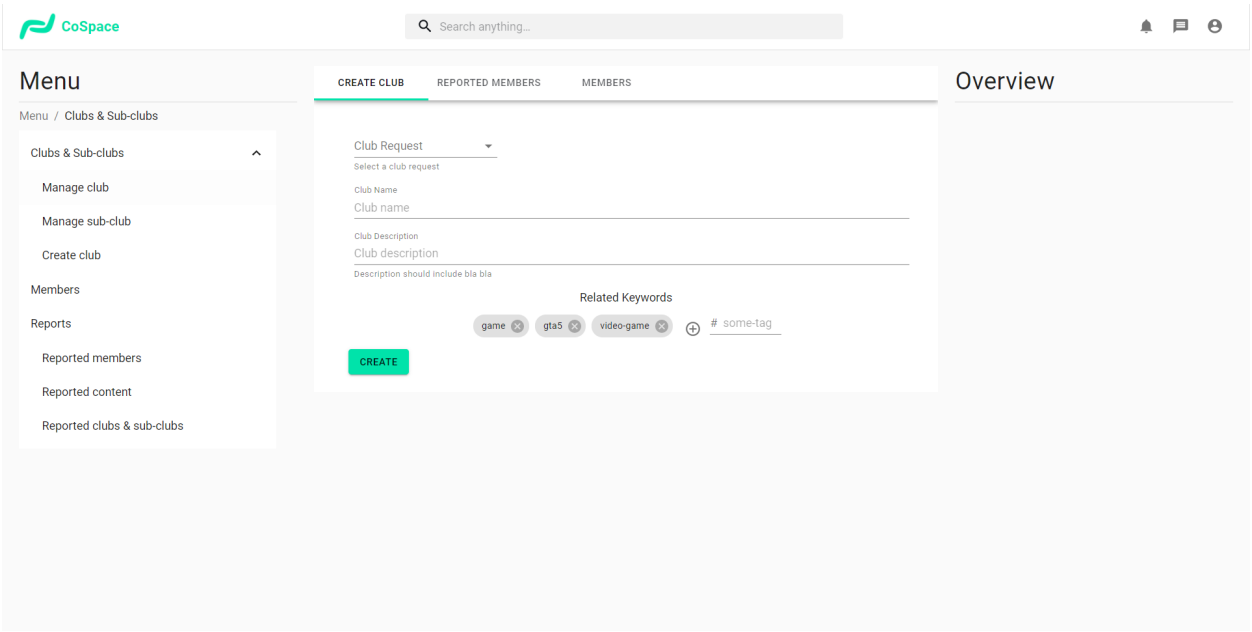


3.4.3 Register & Login





3.4.4 Admin Panel (WIP)



4 Requirements Traceability

In the table below, the use cases are mapped to the classes in the backend service to provide traceability between requirements and the proposed software design. Red font in the use case name means that the design has not been completed for that use case. Yellow font in the use case name means that the design is partially completed for that use case.

Classes/Use Cases (UC)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
JwtTokenFilter	X	X																		
TokenManager	X	X																		
MessageConfig					X															
AdminController													X	X	X	X	X	X		
AuthController	X	X																		
ClubController				X				X	X	X				X		X				
MemberController	X	X			X		X	X	X	X										
PostController				X		X	X													X
SubClubController				X		X	X	X	X	X	X			X		X				
ClubDTO				X				X	X	X				X		X				
LoginRequest																				
MemberDTO	X	X					X	X	X											
PostDTO						X	X													
ReportDTO																				X
ReviewDTO											X									
SubClubDTO				X		X	X	X	X	X	X			X		X				
Authority	X	X				X		X	X	X				X	X	X	X	X	X	X
Club				X					X	X				X		X				
Member					X		X	X	X											
PasswordResetToken																				
Post						X	X													

Question									X				X							
Report																				X
Review											X									
SubClub				X		X	X	X	X	X	X			X		X				
AdminService													X	X	X	X	X	X		
AuthService	X	X																		
ClubService				X				X	X	X				X						
CustomUserDetailsManager	X	X						X	X	X					X					
MailService																				
MemberService	X	X			X		X	X	X	X								X		
PostService						X	X										X			X
SubClubService				X		X	X	X	X	X	X			X						

5 Annexes

There is no separate document as an attachment to this document, as of version 0.1.