HENRI SU - BENOIT LIN

SPORT TOORNAMENT

2024-2025

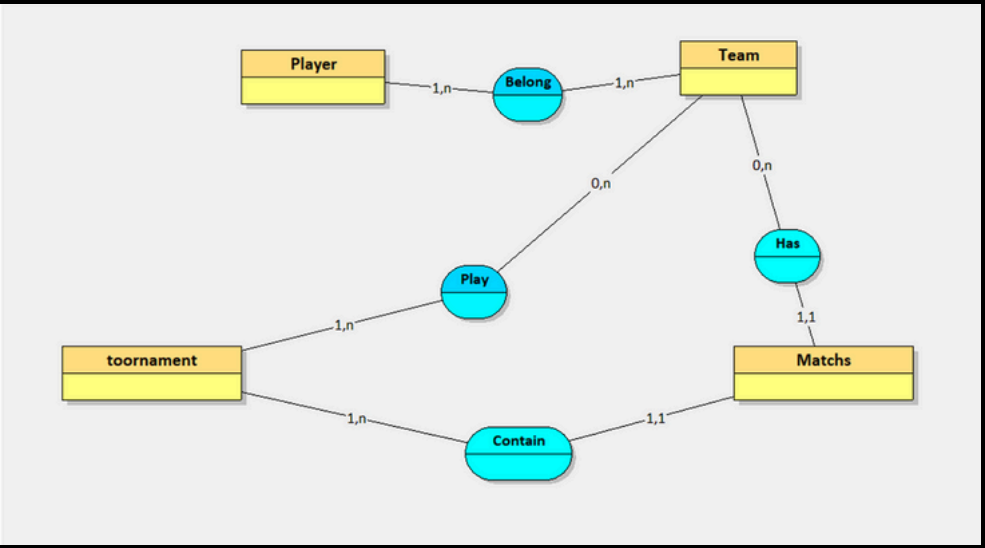
PROJECT

Firstly, we wanted to create a website to propose to everyone from everywhere, a site where he can participate, see new people if you are new from a country and you want to share your passion or if you live in your country and you want to challenge yourself. We searched what we could include in our site and we finally came with this. This project is a database designed to manage a sports system, including players, teams, toornaments, and their interactions. Players belong to teams, teams participate in toornaments, and matches are organized between them. Each entity contains specific information (e.g., players: age, email; teams: coach, captain; tournaments: location, dates). The relationships ensure consistency, such as linking players to their teams and associating teams with tournaments, with unique constraints to prevent duplicates. This system provides a structured way to track and organize sports competitions.

SUMMARY

CHOICE OF THE DATABASE

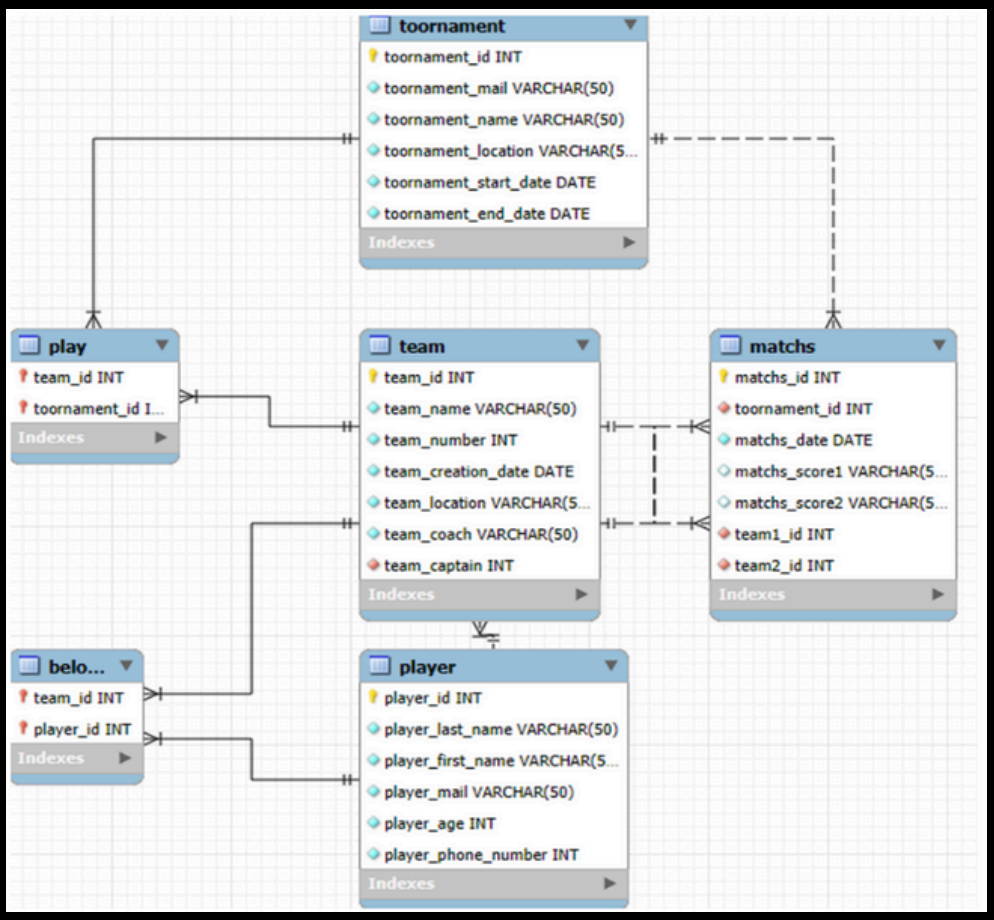
ER DIAGRAM (LOOPING)



We decided firstly to begin with what are the features that we need for our site so we firstly began with toornament, matches, team and players. Then, we had to know how we can link them together so we used, belong, has, play and contain to get our links. Finally, we had to know the relations links, for example, 1 player or more belong to a team but a team is belonged with 1 or more players (a team for a swim toornament only need 1 player but a team for a football toornament needs more).

CHOICE OF THE DATABASE

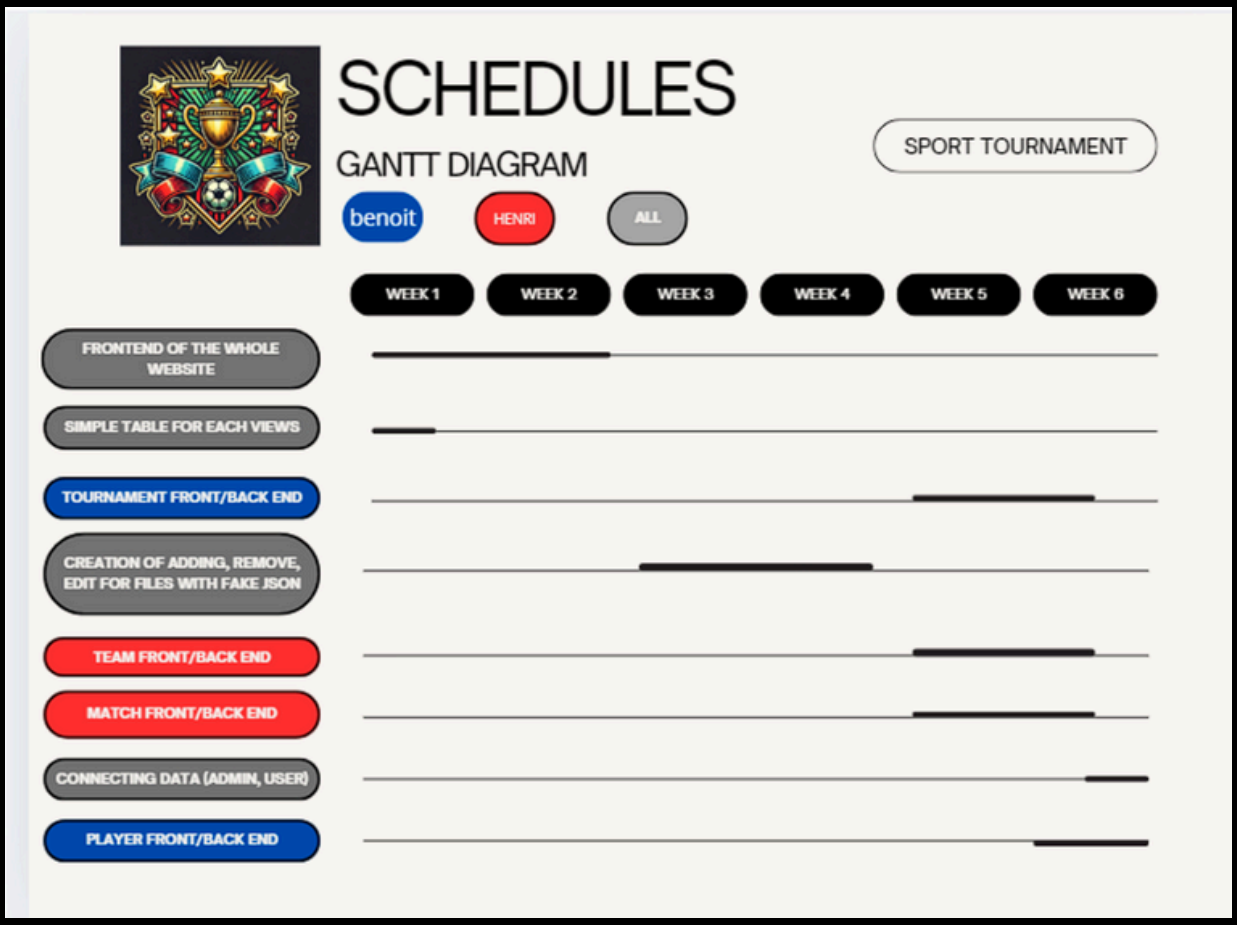
TABLE STRUCTURE DIAGRAM



After the choise of our tables, we had to know what we could put inside, and we decided with the help of our precedent diagram, to put those features for each tables.

SCHEDULE

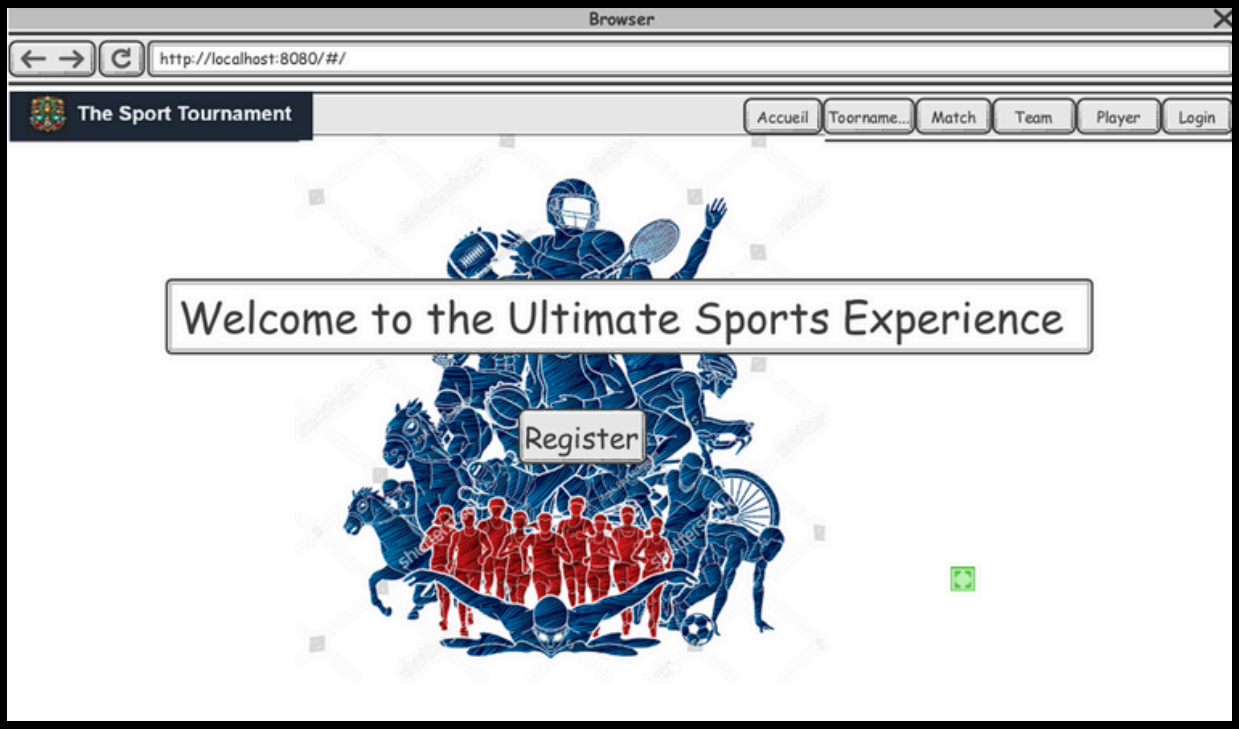
GANTT DIAGRAM



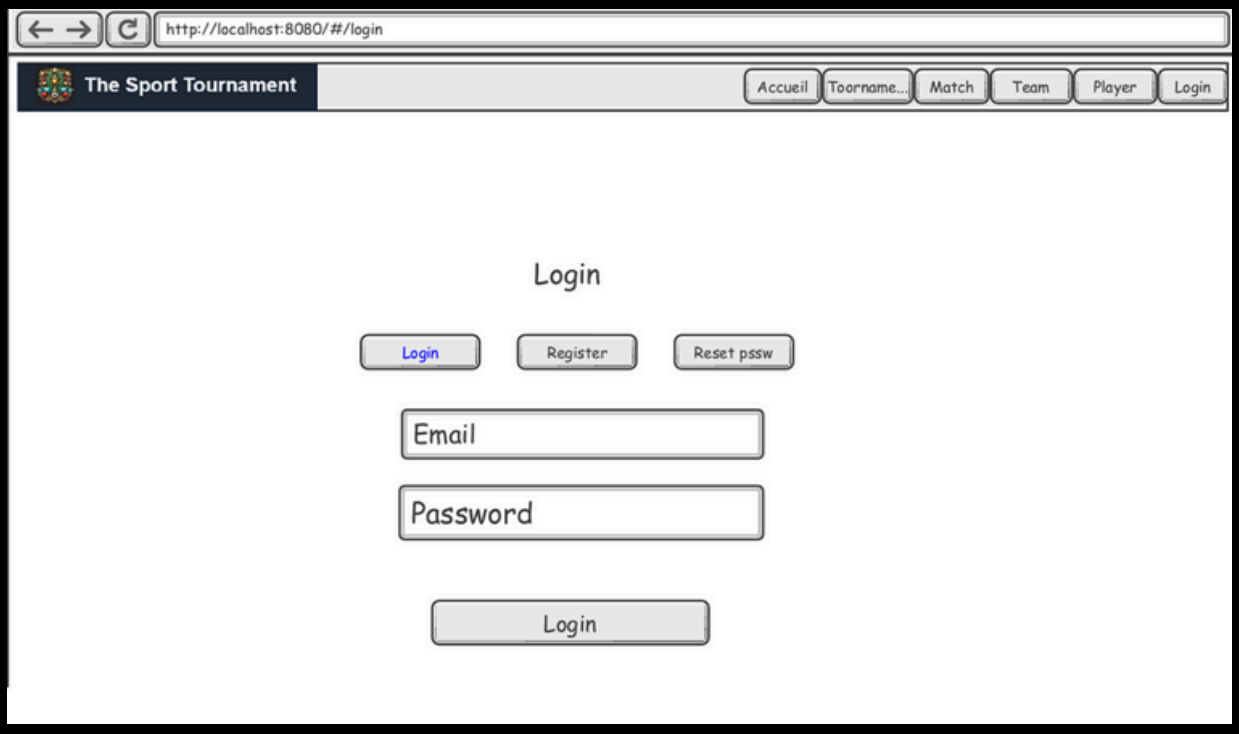
After the choice of how our database would be, we had to decide how our project needs to advance every weeks so we decided to split our project into two groups, the frontend and the backend. For the frontend, we decided to do our site with fake JSON data to have our site ready for a quick presentation and then we only had to change the methods and connect our databse to interact with it. We firstly decided to do half of the pages we need for one person and the other half for the other one and we continually speak together to know if we hava the same vision of our site and correct it like the daily meeting.

WIREFRAMES

WELCOME INTERFACE



LOGIN INTERFACE

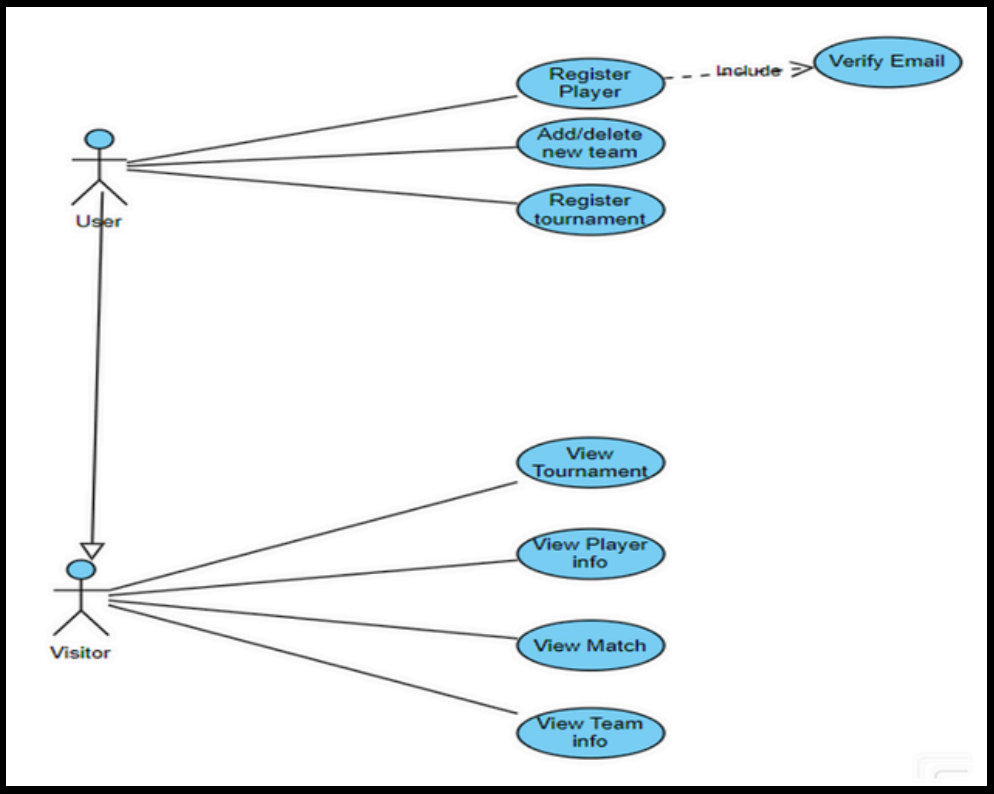


WIREFRAMES

We had to be careful and not immediatly code and then think of how our website should look like. So we decided to do some wireframes to know how our website should look and then we'll code.

USER/ADMIN POSSIBILITY

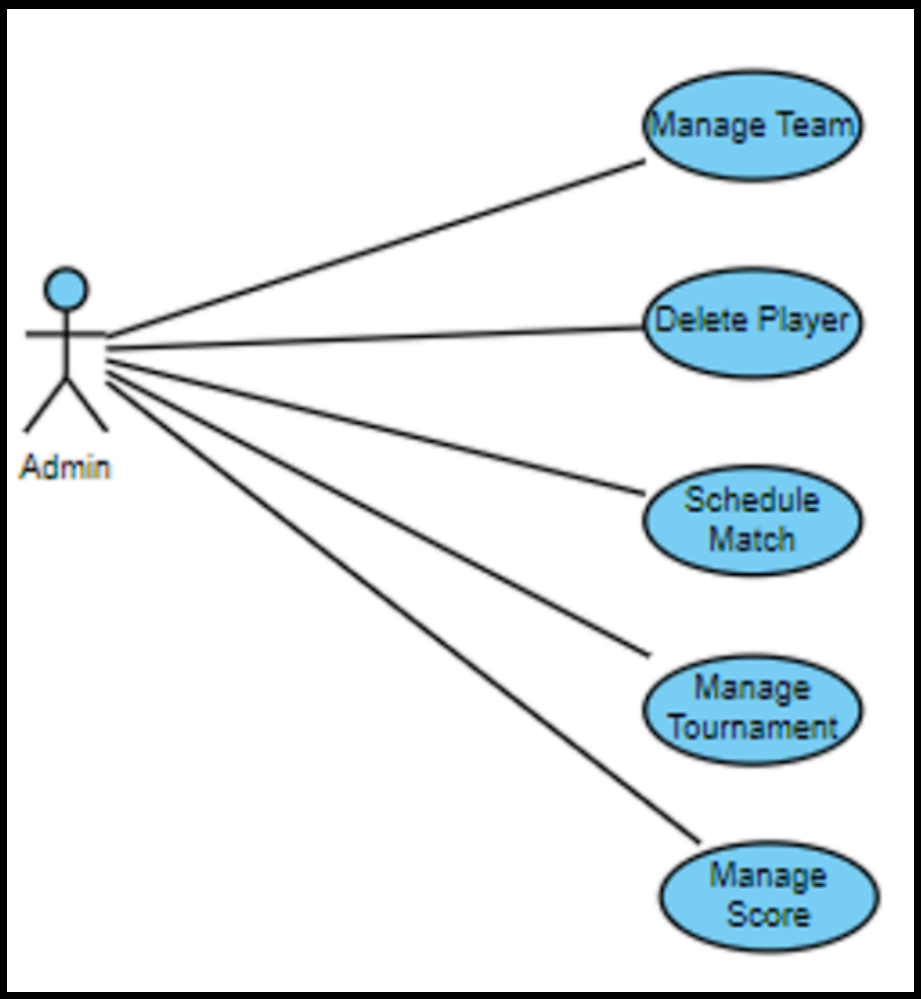
USE CASE : USER



For the different kind of people in our site, we have a visitor but a user is a visitor also with some privileges so the user inherite of visitor.

USER/ADMIN POSSIBILITY

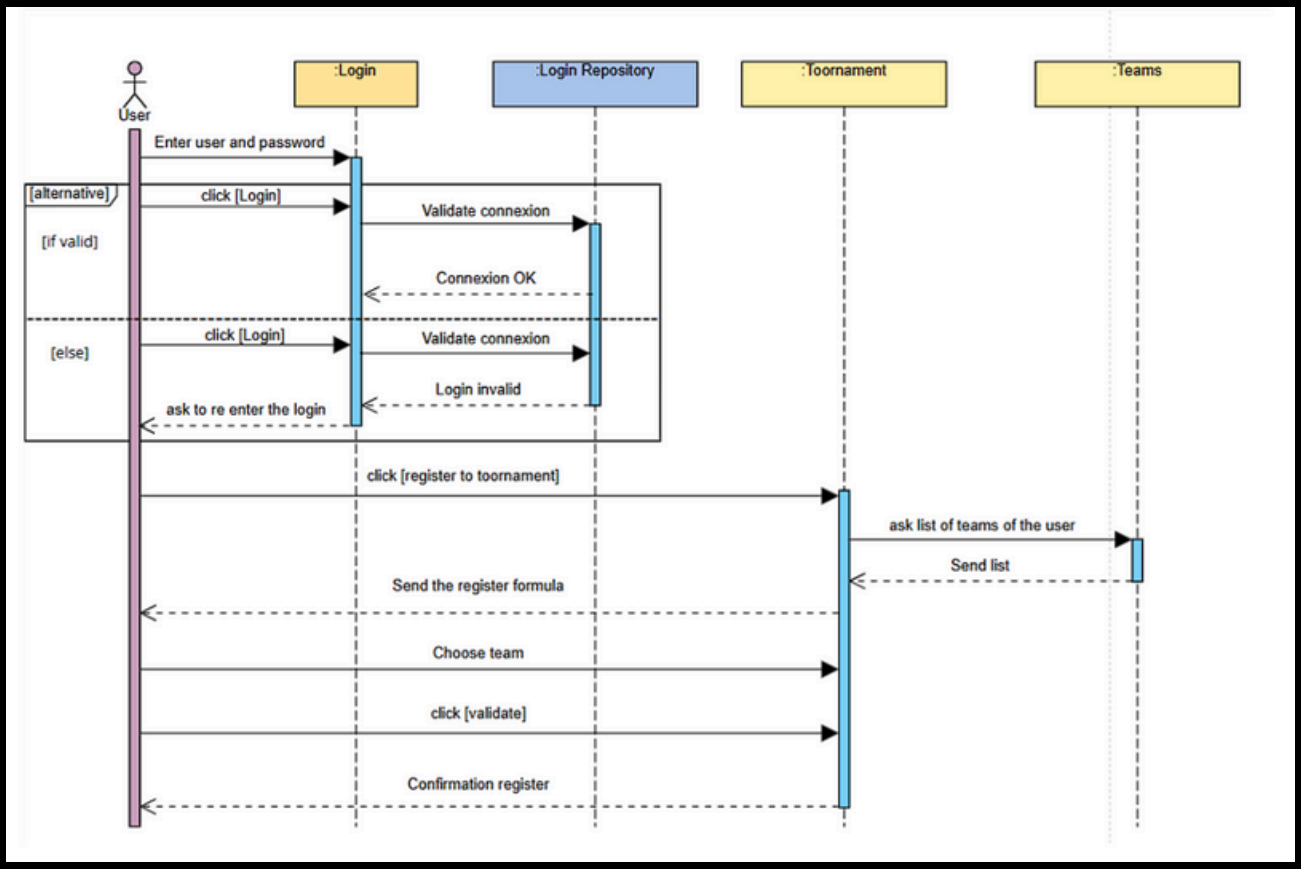
USE CASE : ADMIN



For the admin side, he can do everything.

USER/ADMIN POSSIBILITY

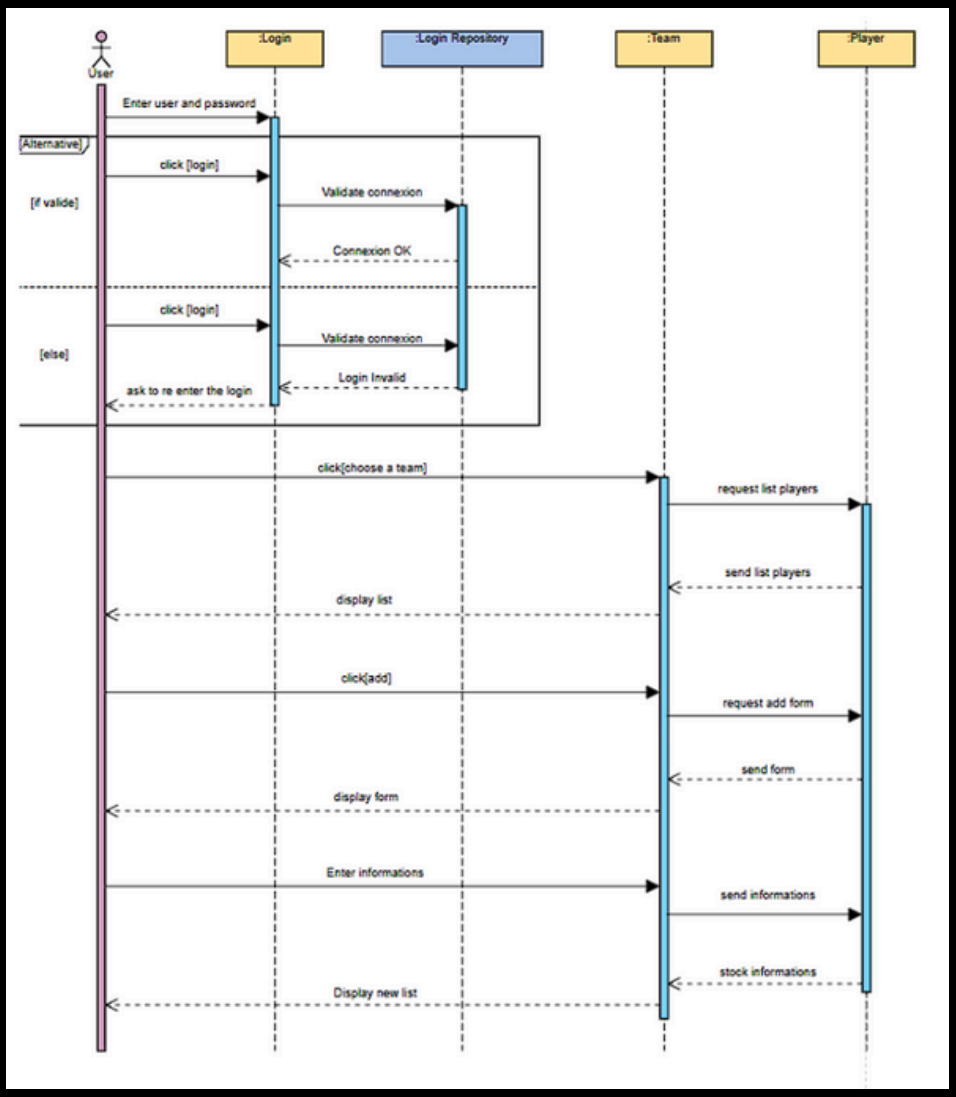
SEQUENCE DIAGRAM (REGISTER CASE)



We also need to know how our site would do when a user wants something. For this case, it's for when he wants to register to a toornament. He firstly needs to connect, and then he needs to click on register and he have to choose which team to add.

USER/ADMIN POSSIBILITY

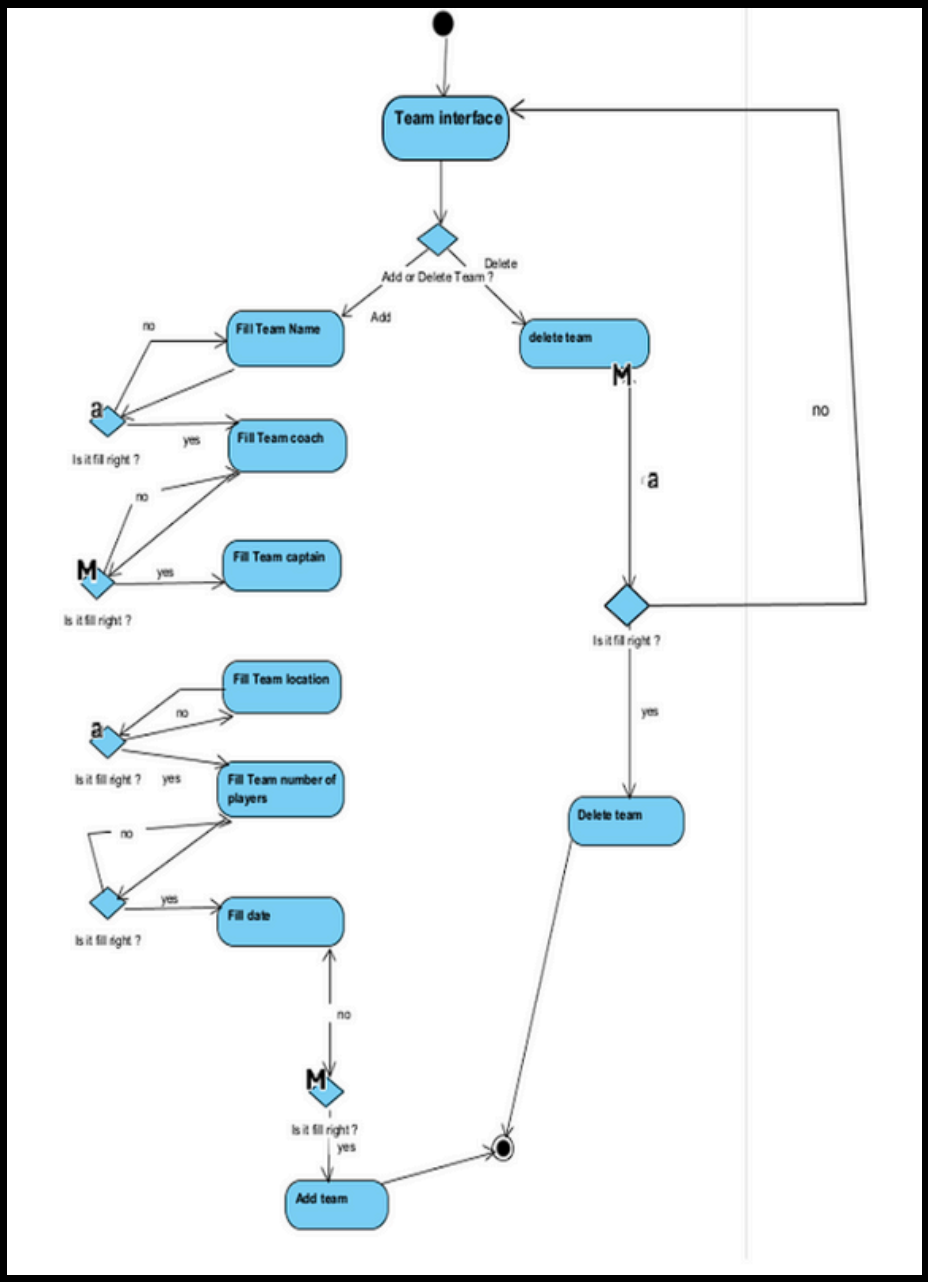
SEQUENCE DIAGRAM (ADD TEAM)



For this case, it’s for when he wants to see add a team. He firstly needs to connect, and then he needs to click on add and a form will appear to let the user enter the informations and send it.

USER/ADMIN POSSIBILITY

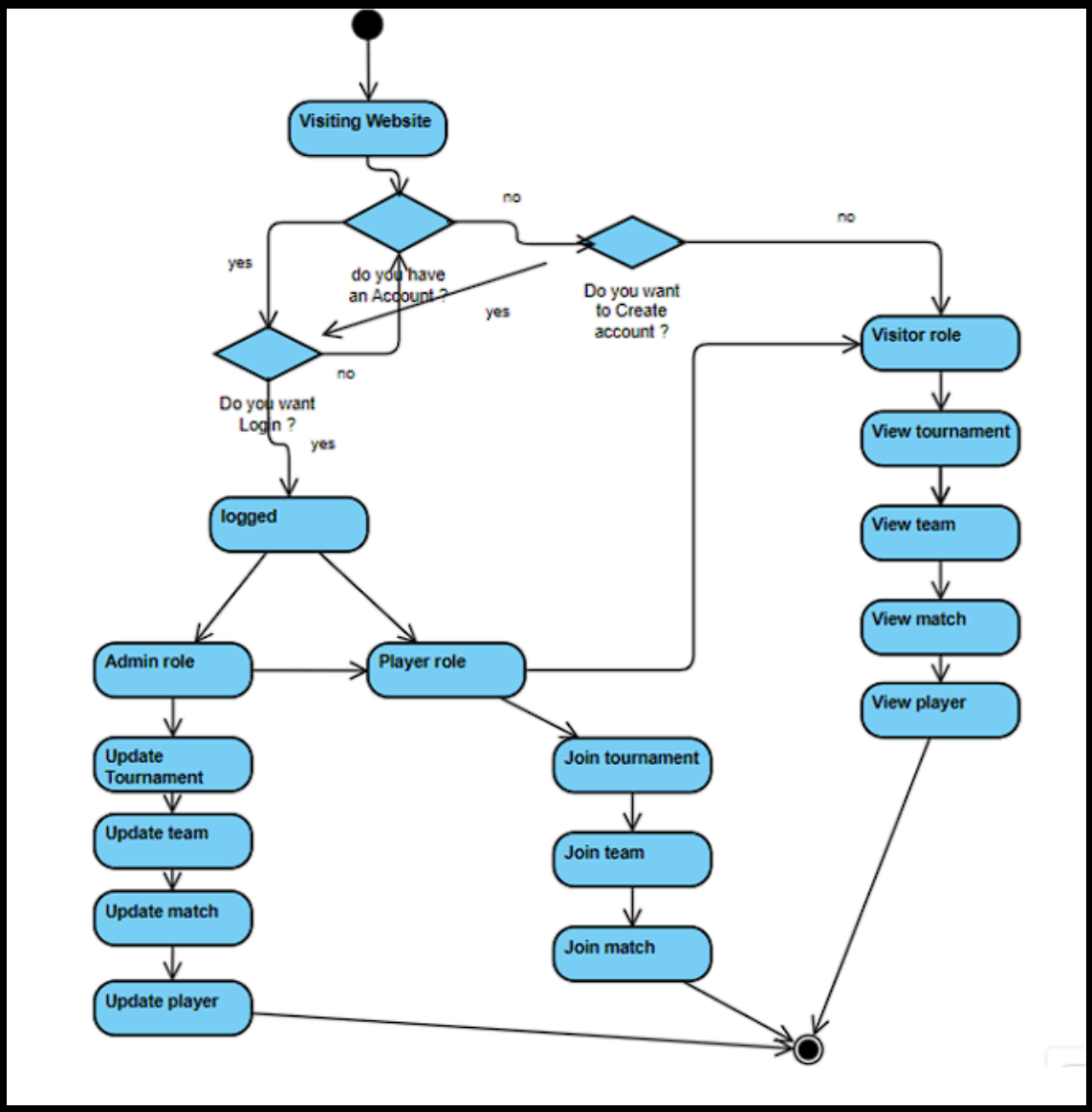
TEAM ACTIVITY DIAGRAM (ADMIN / USER)



We also had to know what is the path for a user or an admin. Each circle is an action and losange is a possible action/question and we show what will happen if it's yes or no. At the end, the point at the bottom is the end point.

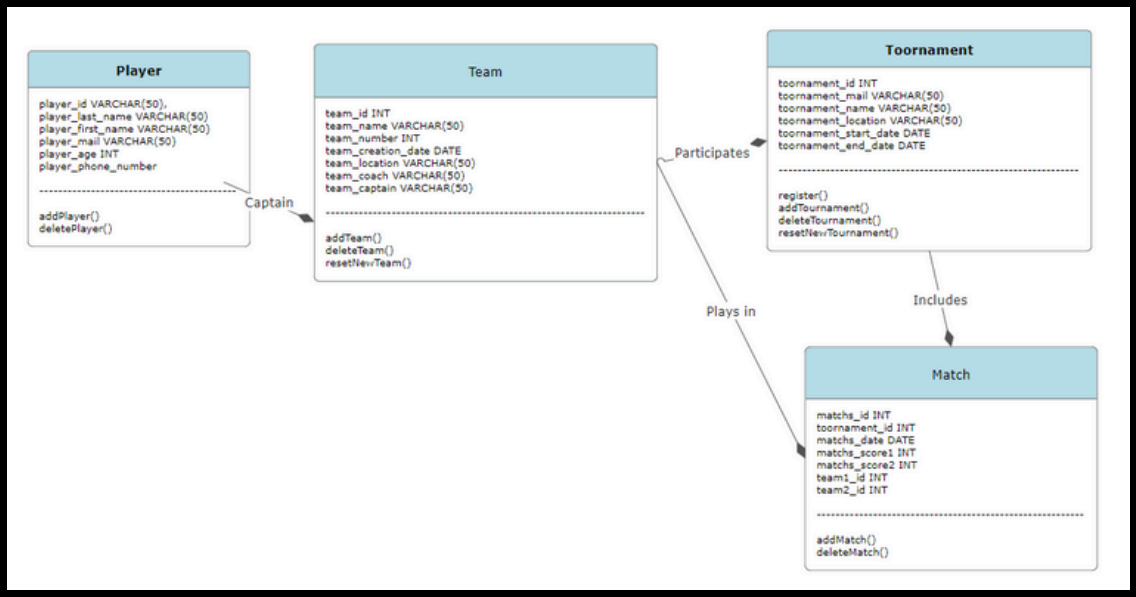
USER/ADMIN POSSIBILITY

GLOBAL ACTIVITY DIAGRAM



USER/ADMIN POSSIBILITY

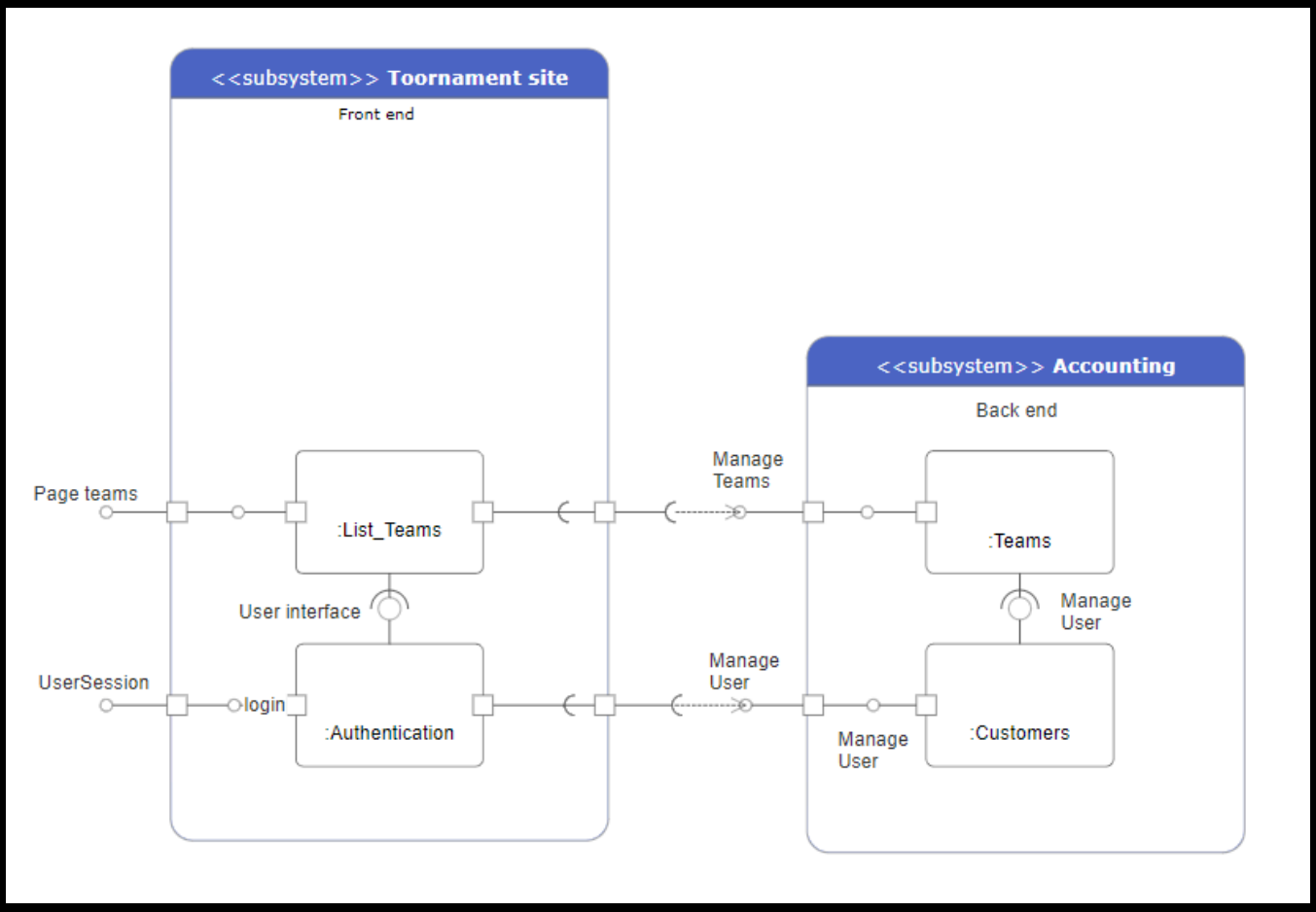
CLASS DIAGRAM



This class diagram models a sports system where players can be captains of teams, teams participate in tournaments, and tournaments host matches. Each class (Player, Team, Toornament, Match) includes relevant attributes, methods, and relationships, forming a structured management system.

USER/ADMIN POSSIBILITY

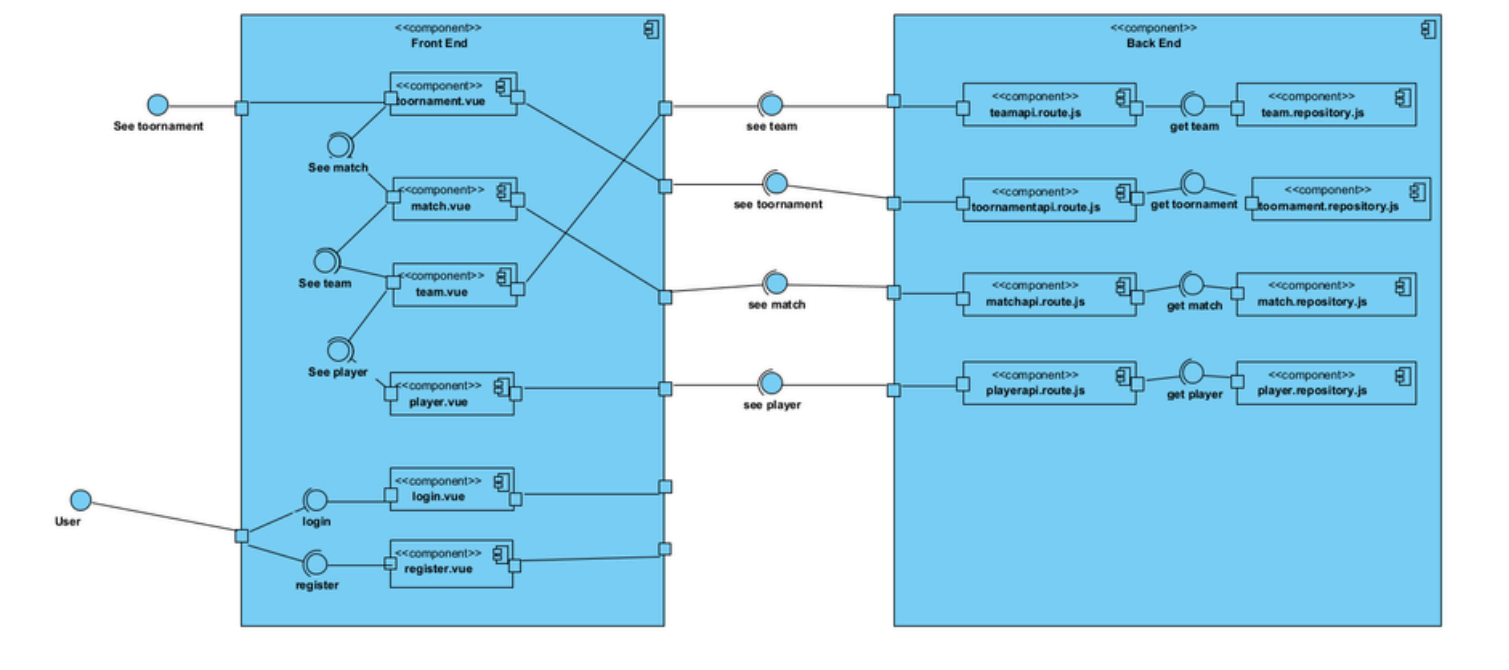
COMPONENT DIAGRAM



The component diagram illustrates a modular software architecture divided into two main subsystems: "Tournament Site" (front-end) and "Accounting" (back-end). This structure ensures a clear separation of responsibilities between user interaction and business logic. Tournament Site (Front-End): This subsystem focuses on user interactions. Accounting (Back-End): This subsystem manages the business logic and data.

USER/ADMIN POSSIBILITY

COMPONENT DIAGRAM



The component diagram demonstrates a client-server architecture, with a clear separation between the Front-End and Back-End subsystems. The front-end is responsible for user interactions and provides the user interface for various functionalities. The back-end is responsible for handling the application's business logic and data management. The front-end components request data or send user actions through specific API routes in the back-end. Each API route processes the request, retrieves the required data from its associated repository, and sends the response back to the front-end.