**MIE350 Team 15 Final Report**

**Team Name:** Lan

**Topic:** Premier League (Soccer) Statistic Analysis

Date: 2022/11/23

**Team members:**

| Name | Role | Email |
| --- | --- | --- |
| Ricky Wu | Back-end Lead | Ruifan.wu@mail.utoronto.ca |
| Jenny You | Front-end Lead | jenny.you@mail.utoronto.ca |
| Richard Jing | Back-end engineer | Richard.jing@mail.utoronto.ca |
| Victoria Li | Research | yitong.li@mail.utoronto.ca |
| Rosalind Wang | UI/UX designer | rosalind.wang@mail.utoronto.ca |
| Eric Wang | Front-end engineer | hbo.wang@mail.utoronto.ca |
| Muchen Liu | Project manager | Muchen.liu@mail.utoronto.ca |

**Table of Contents**

| 1.0 Executive Summary | 2 |
| --- | --- |
| 2.0 Overview of Web App | 3 |
| 3.0 Overall architecture of the system | 3 |
| 4.0 Summary of Requirements | 4 |
| 4.1 Requirement Discussion Methods | 5 |
| 4.2 Functional and Non-Functional Requirement | 6 |
| 4.3 Software Development Methodologies | 8 |
| 4.4 Use Cases | 8 |
| 4.5 Context Diagram | 13 |
| 4.6 Level 0 Data Flow Diagram | 13 |
| 4.7 Level 1 Data Flow Diagram | 13 |
| 5.0 UML Use Case Diagram | 14 |
| 6.0 UML Class Diagram | 15 |
| 7.0 UML State Chart | 16 |
| 8.0 UML Sequence Diagram | 16 |
| 9.0 UML Activity Diagram | 17 |
| 10.0 Access by Wireless Device | 17 |
| 11.0 Quality Assurance Description | 18 |
| 11.1 Testing Procedures for Project | 18 |
| 12.0 Challenges and Difficulties Encountered | 20 |
| 13.0 Lessons Learned | 20 |
|  |  |

**1.0 Executive Summary**

Soccer is a very popular sport all around the world with a considerable amount of soccer fans willing to check accurate game information and discuss it together. The team was activated by the enthusiasm of those soccer fans and decided to design an Online Community Site that is related to Premier League discussions with official accurate information.

The team uses interviews, task analysis, and scenarios to identify the requirements of this website that help design the back end and the front end. In the design process, the team considers both the functional requirements and the non-functional requirements.

For functional requirements, the designed website is supposed to contain the data of both soccer teams and soccer players in the Premier League. To make it convenient for soccer fans, namely the users, to check the data that they look for, the data displayed on this website should have a filter function and the search bar function. Therefore, users can find information about their favourite player or team in a short time rather than browsing through huge amounts of data. Besides, the designed website is expected to have a discussion page that allows users to post their comments and share their opinion when the team or player they are concerned with wins or loses a game. In addition, every user should have an account on this website that keeps their favourite team and player. Hence, the website has both the register and login functions that users can log in to their accounts with their unique usernames and passcodes. After logging in, users can set their favourite team and player on the profile page and check all the discussions posted by them. If users get interested in another team or player, they can also change their profile information or delete previous discussions posted by themselves.

For non-functional requirements, the website can be accessed by wireless devices, such as computers and laptops. For security and safety, the website restricts access to profitable information and protects the private information of users. The website displays statistics clearly and in an organized structure, hence users can easily understand the context. Besides, the website uses visual designs to help users understand and control the system with a 3-second maximum response time.

In this report, the team also displays the structure of the website through the use case diagram, the class diagram, the state chart diagram, the sequence diagram, and the activity diagram. Test cases are designed in the back end to testing the quality and stability of the website. In the end, the team summarizes the difficulties encountered and identifies the lessons learned during the process of achieving this design for further improvement.

**2.0 Overview of Web App**

As the top league of the largest sport [1] on the planet, the Premier League had a great impact around the globe with more than 1.4 billion fans [2]. Among fans of different teams, fierce debates concerning team superiority and game predictions have been raised. Millions of discussions are happening over the internet related to the Premier League; Until now, the Premier League has had 36,486,242 followers on Twitter [3], and those fans read, comment, and post about the Premier League. However, the posts on other websites, such as Twitter [3], might be inaccurate regarding the true performance of players, which can be demonstrated by statistics. The gap is a lack of reliable Online Community Site that supports Premier League discussions with official accurate information.

Therefore, the team is motivated to establish an Online Community Site that is related to Premier League discussions with official accurate information. The goal of this project is to build an Online Community Site; with built-in functions for viewing team statistics, posting discussions, and linking statistics to comments.

The stakeholders of this Community Site are other football-related websites, such as Twitter [3], Canadian Soccer news [4], etc., and the intended users. The key users of this Community Site are soccer fans who want to discuss soccer games, clubs, and players together using this soccer-related app/website, especially for Premier League. However, other websites mentioned above, only have one of the functions of checking official information or discussing, not both. The information gap between users may lead to unnecessary arguments, and the lack of discussion function makes it hard for users to enjoy the games with others. Based on this situation, the team created this site with both functions that will provide a friendly environment for fact-based discussion for the users.

**3.0 Overall architecture of the system**

The framework that is going to be used for the web application is Spring Boot. It is the starter of the java-based REST server, and the H2 database is going to be implemented together to help with managing the data storage of the application. JavaScript-based front-end user interfaces will be developed to present the content to the users.

The major CMS model entities are users, player statistics, team statistics, and forum posts; datasets store information about these entities. As for users, usernames, profile information, their favourite soccer teams, as well as their historical posts are stored in the database. Statistics entities include soccer teams, players, and match results; datasets containing their information are uploaded into the system. The dataset for the forum will record the content of the discussions, and the id of the user who posts the discussion.

Correspondingly, webpages will be presented to users and enable user operations, and some involve control of the database and display of the information. As the basis of the web application, soccer teams, and players will be listed on a page where a search function will be implemented. Users will also be able to use the search function to filter out required outcomes. Besides, users can add soccer teams and/or players to their favourite list.

Users can post discussions on the forum page. Once a discussion is compiled and posted, a new entry will be created into the forum database. Meanwhile, the post will become visible to all other users on the page. Users can also post discussions as comments to other posts.

Lastly, the user homepage shall display information for that user only. Aside from basic user profile information, soccer team, player and discussion entities are referenced to users as described in the database structure. Users will be able to view their favourite team and player and historical posts on the user page. This is also where users can operate with the list and posts, example functions are reading and deleting. (Figure 1)

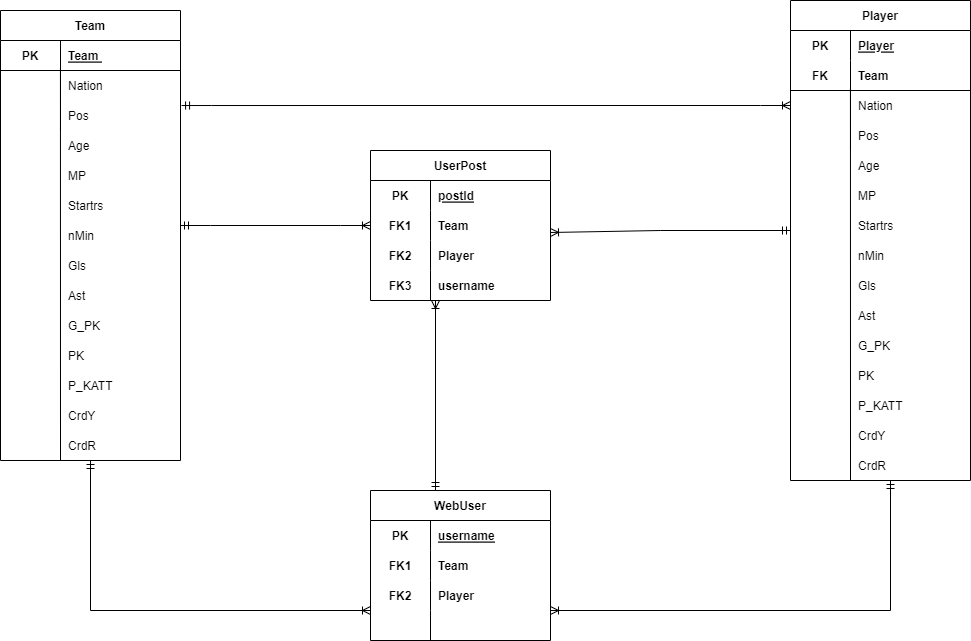


Figure 1: E-R Diagram

**4.0 Summary of Requirements**

The team used three methods, which are interview, task analysis and scenarios (Table 1) to discover functional and non-functional requirements for the system to have an accurate understanding of the system and how we build it. In addition, the team determined software development methodologies, use cases, context diagrams, data flow diagrams as well as the architecture of the system.

4.1 Requirement Discussion Methods

Table 1. Method used for decision making on requirements

| Methods | Details |
| --- | --- |
| Interview | - Who to interview   * Users (Key stakeholders) who have used soccer-related apps/websites to search for soccer-related statistics or discussions   - Designing interview questions   * Which apps/websites are you currently using for searching or discussing soccer-related statistics? * How often do you use the app/website? * What feature/function is your favourite? Why? * What features/functions do you dislike? Why?   - Preparation   * Interviewees are selected if they have used soccer-related apps/websites to search for soccer-related statistics or discussions * Introduce the context and motivation of our interview to our interviewees. Inform the interviewees that our interview is anonymous and please provide true experience. Then, ask for consent * Set up online meetings through zoom   - Conducting the interview  - Follow-up |
| Task analysis | The task is “Discuss the performance of a specific soccer club, ex. Using Twitter”.  Subtasks:   * Search for the tag of this soccer club in the search bar * Read through the post and see someone that holds a different opinion about the team's performance on one game * Open a new page on the web browser and search for the statistics of this team * Find a website that has game statistics * Read through the information on the website and find the soccer club in the discussion * Go back to the Twitter post and comment/post with the statistics found * In our system, we wish to achieve the same task with less subtasks: * From the main page, click on the club * Read through the post and see someone that holds a different opinion about the team's performance on one game * Click on the tag of this team and look at the statistics * Go back to the post and comment/post with the statistics found |
| Scenario | - Add tags to posts   * Starting point: The user posts a discussion on player/game stats * The user wants to justify his points and arguments * Then the user could use tags to refer to the player/game being discussed * Tags are linked to web pages of that player/game being referenced * Once clicked the tags, the webpage shall be redirected to the stats page * Other forum users will also be able to see and click on the tags * After viewing the stats, the user may go back to the post (forum page) with the back button   - Find and delete historical posts/reposts/comments on homepage   * Starting point： Users posted a discussion/comment in the forum * Later, users want to delete the post but cannot find it in the forum page * Users navigate to personal homepages * Scroll down and locate the historical posts section * The section contains only historical posts made by the user himself * Posts are arranged based on date and time * Users locate the post more easily than in the forum * Users can click on the post to navigate to the forum page * Users can delete the post but cannot make any other changes to the post   - Check Player/team statistics   * Starting point: The user wants to find players'/Teams' statistics * The users want to check specific players/teams since they want to make analysis on them * Users navigate to the statistics page * The page has filters with features of players/teams * Users could select filters on the page with their interests * The section contains only certain historical data selected by users |

4.2 Functional and Non-Functional Requirement

Based on analyzing results from the 3 requirement discussion methods, the team went through four steps to finalize requirements. The four steps include requirements elicitation, requirements specification, requirements validation, and requirements negotiation, and finally classified requirements into two classes - functional requirement (Table 2) and non-functional requirement (Table 3).

Table 2. List of Functional Requirement

| Type | Function | Requirement |
| --- | --- | --- |
| Functional Requirement | User Homepage | * Should let users create accounts & store account information (account number, username, password) * Users should be able to access their own User profile (username, portrait, account number……) * Users should be able to edit their favorite player/team * Users should be able to view their favorite player/team * User should be able to post personal Forum posts/delete post/ comments |
| Public Discussion Posts | * Users should be able to post comments * Users should be able to read posts from others |
| Team &  Player Stats | * Users should be able to select a team and discover information about that team * Users should be able to check team/players’ statistics * Users should be able to check information with filters |

Table 3. List of Non-functional Requirement

| Type | Requirements | Description |
| --- | --- | --- |
| Non-functional Requirement | Accessibility | * Be accessible from different wireless devices |
| Security | * Restrict access to team and player statistics to ensure accuracy |
| Safety | * Users can only edit their own information by entering the correct credential on the login page |
| Display | * Display statistics clearly and in an organized structure, so users can easily understand the context * Use visual designs to help users understand and control the system |
| Speed | * Have a 3-second maximum response time |

4.3 Software Development Methodologies

The team used waterfall development to obtain a smooth and coherent workflow. Considering the short period, we were able to have each stage down with efficiency and correctness, and we were able to stick to the requirements throughout the process.

Since the team did not have enough capability or knowledge about website development at the beginning of the development process, the team had to respond to changes in our initial project plan and design. The team provides rapid feedback to each other during weekly team meetings and makes changes incrementally throughout the design process. In addition, the team updated the project log regularly after each weekly team meeting to make sure the team finished each stage on time and check if the design meets our functional and non-functional requirements. For every stage of the project, the team takes into consideration the target users to think about fulfilling the goals and needs of the users.

Besides, in order to ensure the quality and simplicity of the system, the team used pair programming strategies to share the ownership of the code. Everyone on the team is able to access the codes to make corrections and examine the core functionalities and simplicity of the current system.

Overall, The team took advantage of the benefits of waterfall development methodologies throughout the software development.

4.4 Use Cases

To fulfill the requirements of allowing users to view and/or edit their player/team favorite list, and view players’ and teams’ statistics, the team chooses to have use case 1.This is the basis and is essential for our forum.

Table 4. UseCase 1

| Use Case Name: View statistics of players and teams, and label favorite players and teams. | ID:1 | Important Level: High | |
| --- | --- | --- | --- |
| Primary Actor: Soccer fans | | | |
| Brief Description: This use case describes how we view statistics based on different teams and players, and mark the favorite teams and players | | | |
| Trigger: User chooses favorite teams and players to keep track of their statistics  Type: External | | | |
| Normal Flow of Events:   * Users open the statistics page and find the team icon. * Users view the team statistics and match histories. * Users mark the team they selected as their favorite team * Users find the desired team player icon. * Users view the player statistics. | | Information for Steps:  -> team introduction  -> team historical match statistics  -> list of team players    <- player name  -> player statistics | |
| Sub Flows:  S-1: Mark the users’ favorite team  1. System receives the desired team name from the users  2. Users click on the marking sign  3. System creates a team icon on user home page and puts  the team icon at the top of all teams | | Information for Steps:    <- team name  -> favorite team name  -> order of the team icons on the users’ page | |
| Alternate/Exceptional Flows:  E-1: No match  1. The website proposes some alternative teams/players based  on the input name  2. Users choose one of the teams/players or decide not to  choose teams/players | | Information for Steps:    -> potential teams/players    <- selected teams/players | |
| Summary of Inputs | | Summary of Outputs: | |
| Description | Source | Description | Source |
| Favorite players’/teams’ names | Users | Team historical match statistics | Website pages |
| Team name | Users | Team introduction | Website pages |
| Player name | Users | List of team favourite players | Users |
|  |  | Player statistics | Website pages |
|  |  | Favorite team name | Users |
|  |  | Order of the team icons on the users’ page | Users |

The choice of use case 2 is to fulfill the requirement that allows users to read and/or post posts. This is an important requirement for our forum, and this is the reason that the team chooses this use case.

Table 5. UseCase 2

| Use Case Name: Read and Post Discussion/Comment on Discussion board | ID: 2 | Important Level: High | |
| --- | --- | --- | --- |
| Primary Actor: Soccer Fans | | | |
| Brief Description: This use case describes how users post discussions and add tags while posting. | | | |
| Trigger: Soccer fans want to discuss soccer games, soccer teams, and players easily and conveniently.  Type: External | | | |
| Normal Flow of Events:   * Users open the forum and read others’ discussions about soccer games. * Users click on posting/comment and editing their own discussion/comments with context and tags. * if not finished editing, perform S-1. * Users can select whether to post the discussions/comments to the public after editing. * if cancel posting, perform sub flow S-2. * if make further editions, perform sub flow S-1. * Users post the discussion/comments successfully | | Information for Steps:  -> Existing posts on the forum    <- Users key in contents      <- Decisions Users make        <- Users’ discussion posts | |
| Sub Flows:  S-1: Update edition  1. Save current editing  2. Users click on postings on discussion boards/comments to continue editing previous discussions/comments that have not been posted.  S-2 Cancel posting  1. Users cancel posting the discussion/comments to the public, return to S-1.  2. Users decide to not save the post and exit the posting function. | | Information for Steps:    -> Saving confirmation  <- Users continue edition        -> Canceled posting  -> Cancelation final confirmation of posting | |
| Alternate/Exceptional Flows:  E-1: Upload Failed  1. The website shows the failure message.  2. Back to S-1 | | Information for Steps:    -> Failing message "Upload failed".  -> Postings saved | |
| Summary of Inputs | | Summary of Outputs: | |
| Description | Source | Description | Source |
| Contents (discussions/comments) the users key in | Users | Existing posts on the forum | Website |
| Decisions users make of positing or not | Users | Saving/cancelation confirmation | Website |
| Continuing edition users make | Users |  |  |
| Discussions users post | Users |  |  |

Choose use case 3 is to fulfill the requirement of allowing users to delete and update posts on their profile page.

Table 5. UseCase 3

| Use Case Name: Delete and update the discussion from the user homepage | ID: 3 | Important Level: High | |
| --- | --- | --- | --- |
| Primary Actor: Soccer fans | | | |
| Brief Description: This use case describes how users to update and delete their historical posts from the user homepages | | | |
| Trigger: The soccer fans wish to withdraw or edit former opinions already posted on the discussion board from the user homepages | | | |
| Normal Flow of Events:   * Users open their user homepages. * Users select the previous postings * Users scroll down to the certain post they select. * Users click on the “Manage” option of the post. * Users can edit/delete the previous post * Users can give final confirmation of whether they perform the deletion and edition. * if cancel action, perform sub flow S-1. | | Information for Steps:  -> Users’ home page  -> Users’ personal postings  -> Users’ choice of the specific post  <- Users’ decision of management  <- Users make decisions  <- Users’ final confirmation | |
| Sub Flows:  S-1: Undo the edition/ deletion  1. Users click on the return option and go back to the homepages  2. Edition/deletion failed | | Information for Steps:    <- Users make decisions    -> Actions canceled | |
| Alternate/Exceptional Flows:  E-1: Deletion Failed  1. Website shows the failing message.  2. Back to Normal Flow step 5 | | Information for Steps:    -> Failing message "Request failed".  <- Users make decisions | |
| Summary of Inputs | | Summary of Outputs: | |
| Description | Source | Description | Source |
| Users’ decision of management | Users | Users’ home page | Website |
| Decisions of edit/delete users make | Users | Users’ personal postings | Website |
| Deleting/updating confirmation | Users | Users’ choice of the specific post | Users |
| Users’ decision of returning to the homepage | Users | Actions canceled | Users |
|  |  | Failing message "Request failed". | Website |

4.5 Context Diagram (Figure 2)

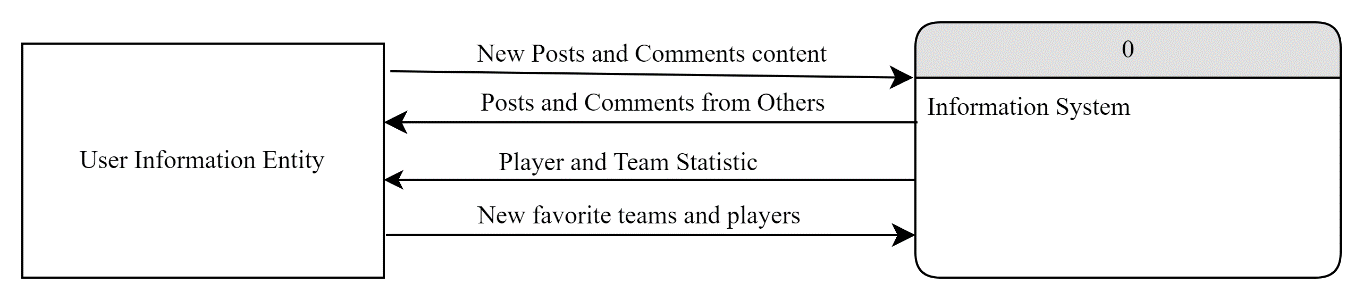


Figure 2: Context Diagram

4.6 Level 0 Data Flow Diagram (Figure 3)

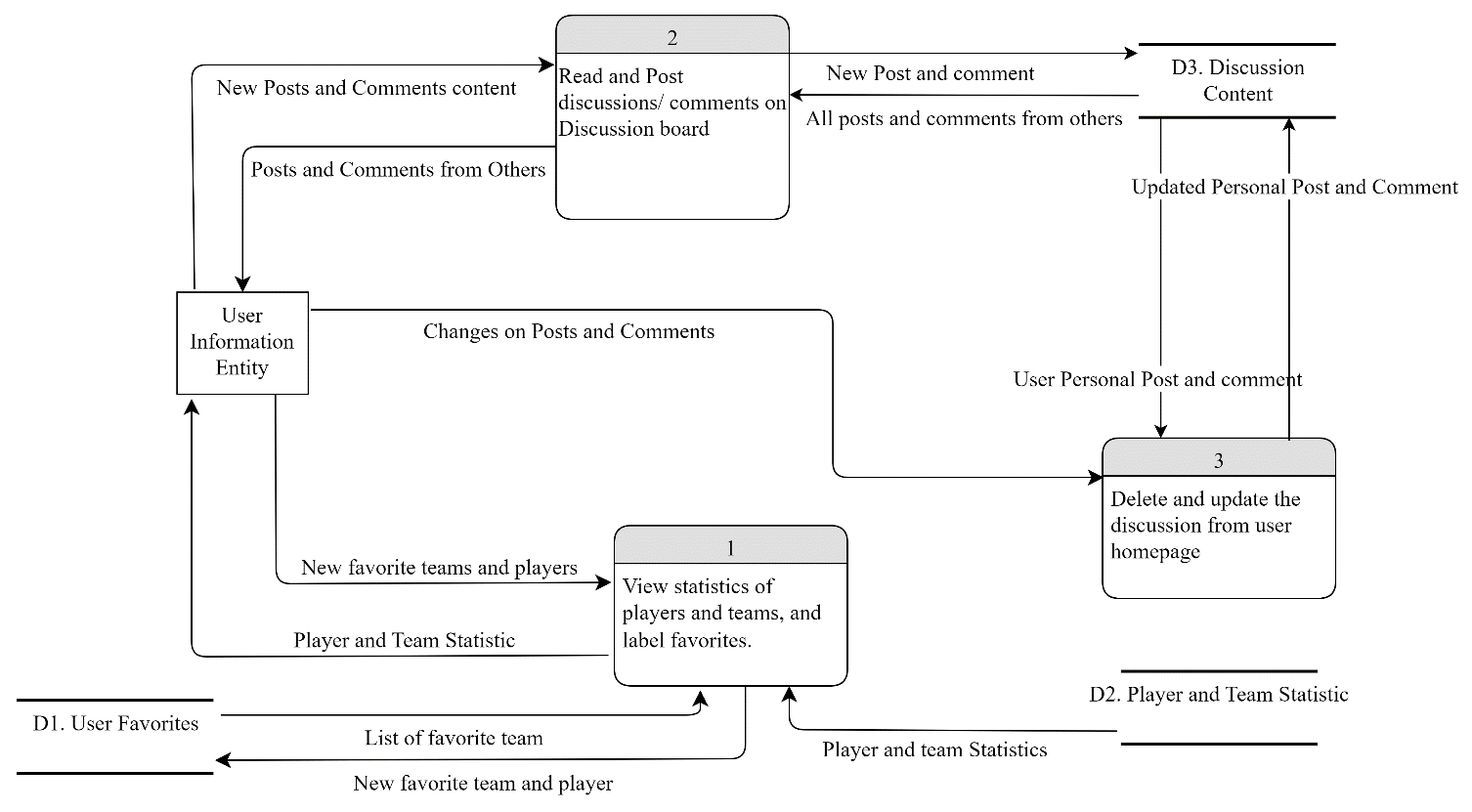


Figure 3: Level 0 Data Flow Diagram

4.7 Level 1 Data Flow Diagram

* Level 1 Data Flow Diagram Use Case 1 (Figure 4)

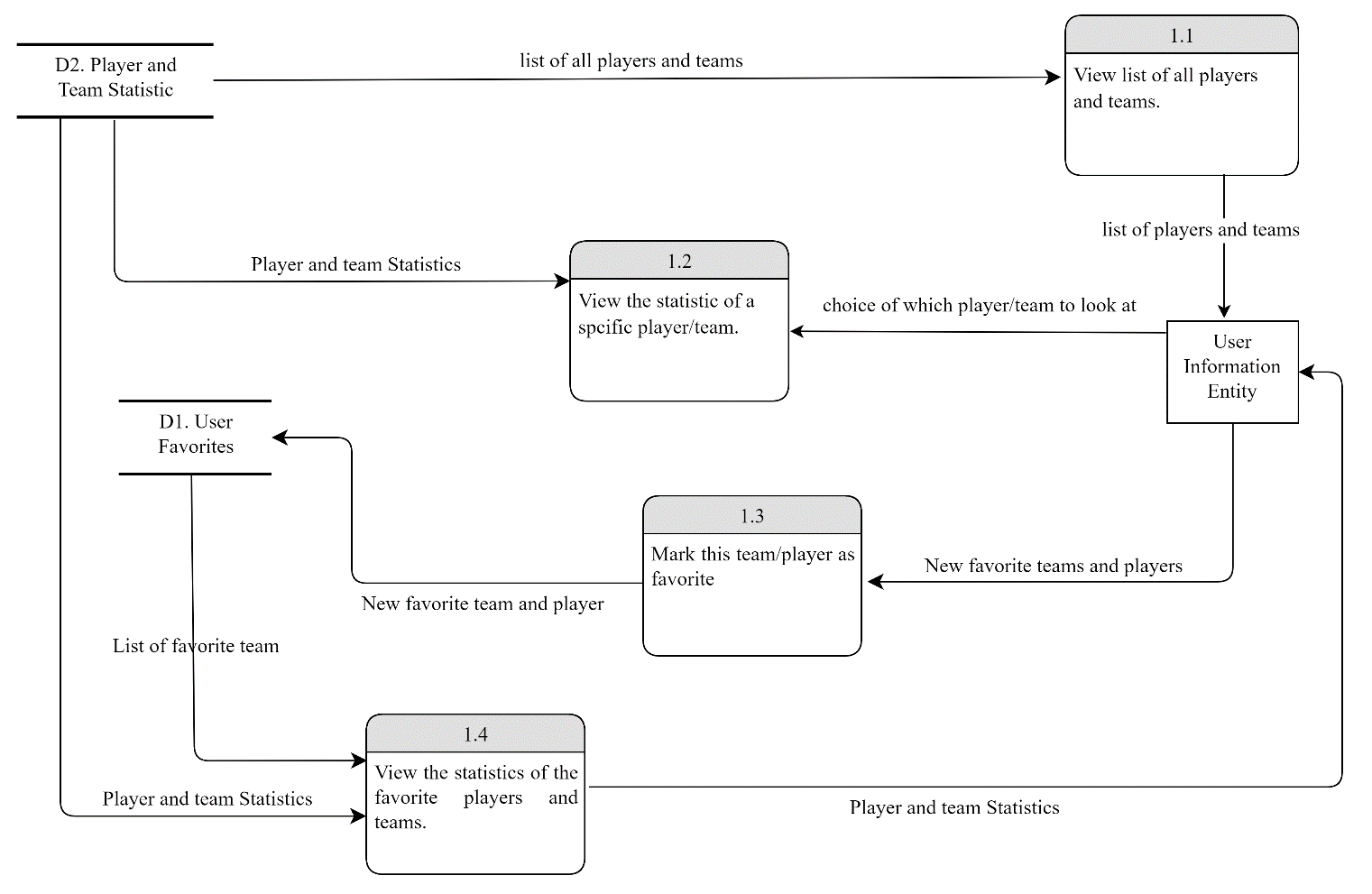


Figure 4: Level 1 Data Flow Diagram (Use Case 1)

**5.0 UML Use Case Diagram** (Figure 5)

The use case diagram shows the actor(soccer fans), collection of use cases, and associations of the use cases of the system.

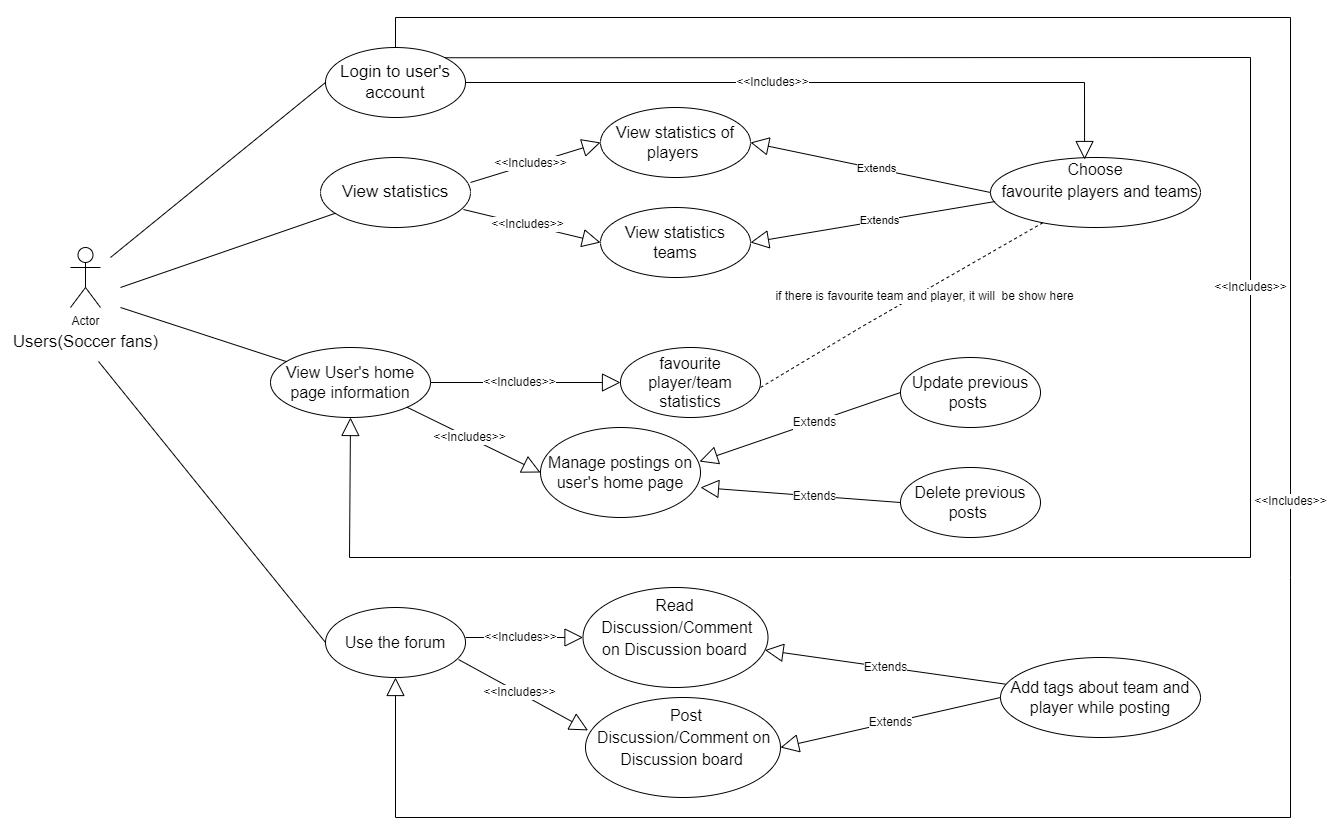
****

Figure 5: UML Use Case Diagram

**6.0 UML Class Diagram** (Figure 6)

The class diagram shows the 4 classes in the object oriented system and how the classes interact with each other. The attributes and type of variables are determined based on the entities and the methods include all the methods on the interfaces.

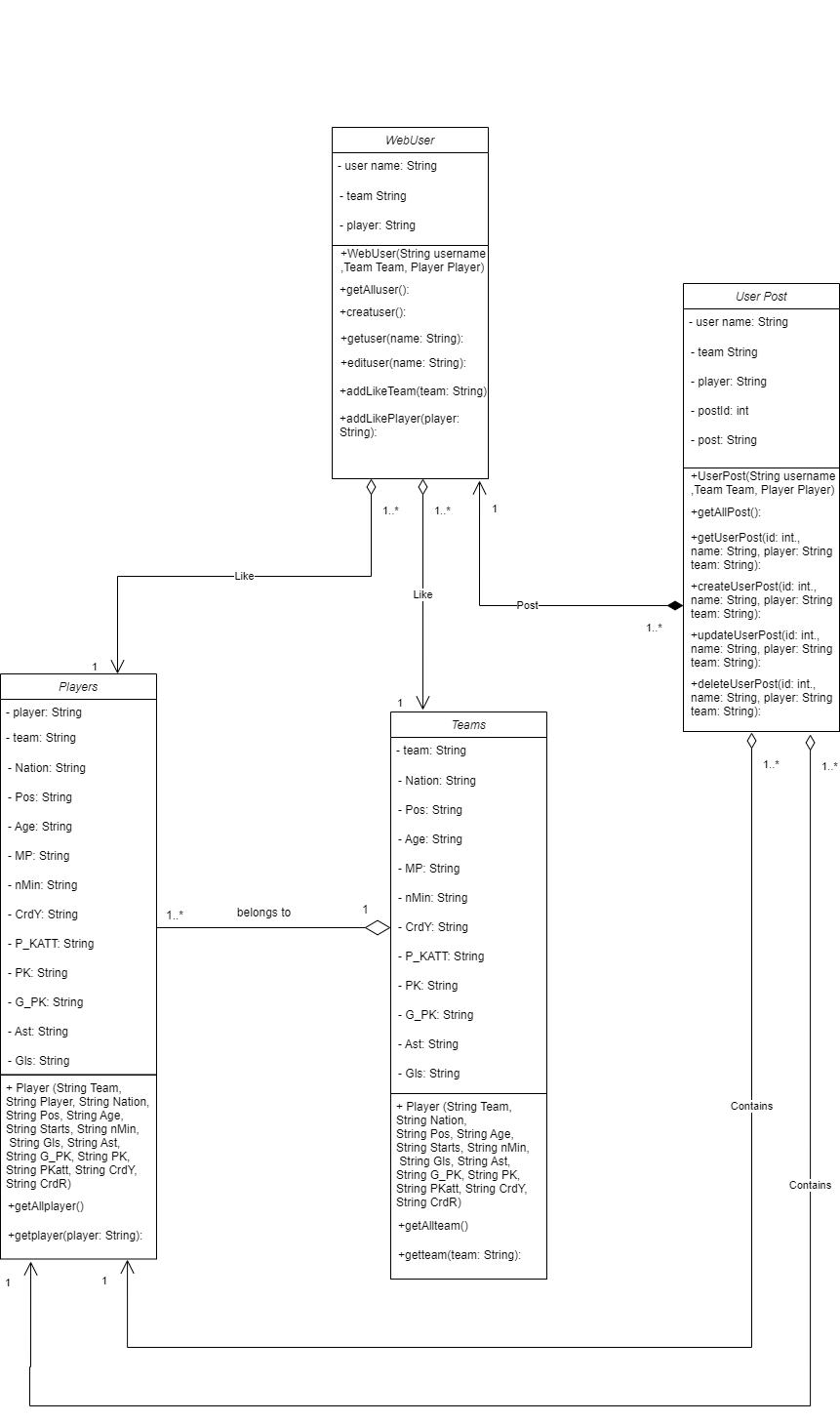
****

Figure 6: UML Class Diagram

**7.0 UML State Chart** (Figure 7)

The UML statechart diagram describes the states that a user goes through when a user creates a discussion on the forum.

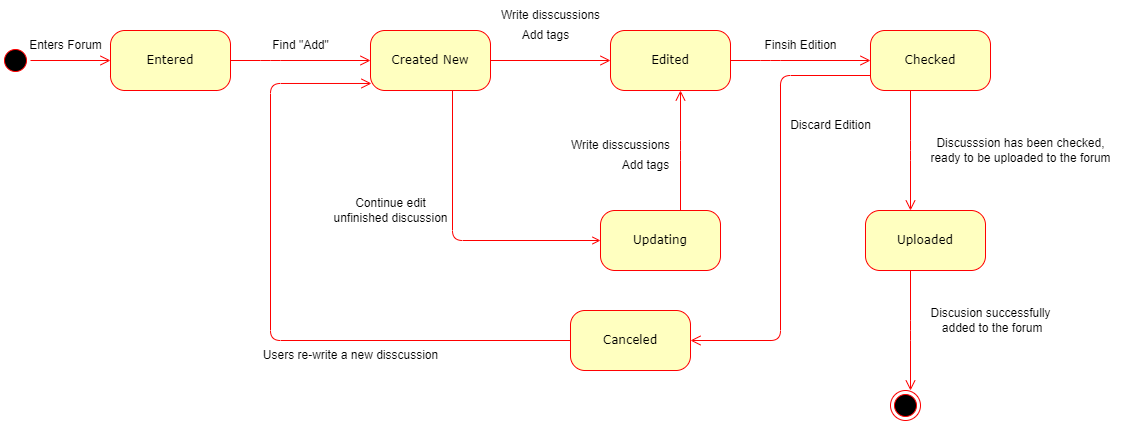


Figure 7: UML StateChart(create discussion)

**8.0 UML Sequence Diagram** (Figure 8)

The UML Sequence Diagram models the control pattern of the first user case about Viewing statistics of players and teams, and labels favorite players and teams.

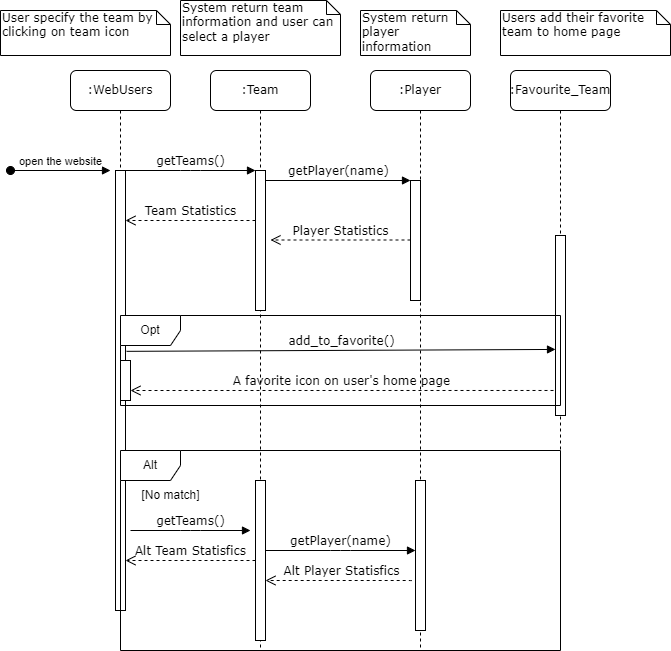


Figure 8: UML Sequence Diagram

**9.0 UML Activity Diagram** (Figure 9)

The UML Activity Diagram is based on Use Case 1(Table 4) and Level 1 Data Flow Diagram (Figure 4).

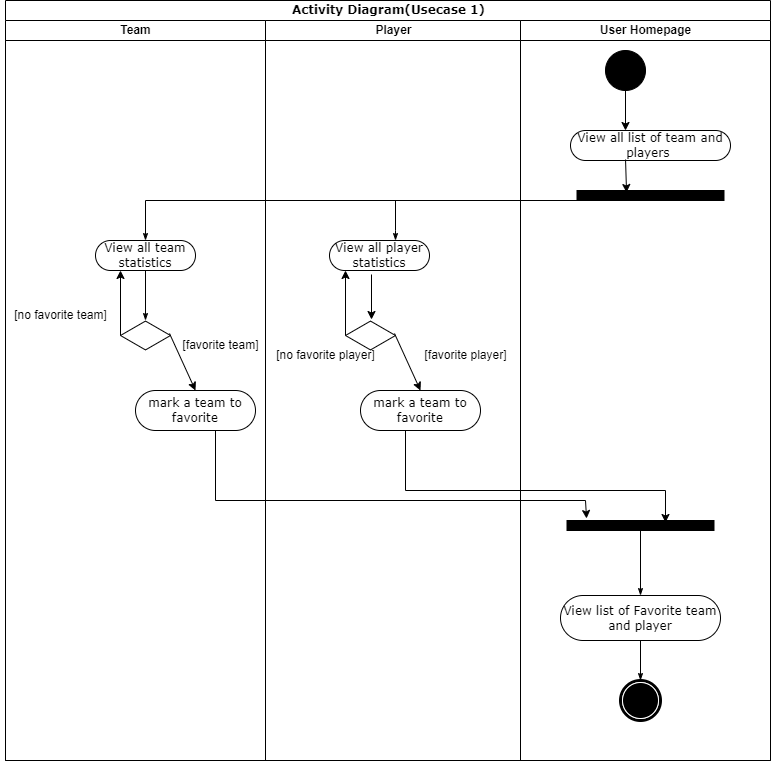


Figure 9: UML Activity Diagram

**10.0 Access by Wireless Device**

Currently, the web application can be run on computers and laptops with Intellij and Node.js installed. Intellij supports backend CMS running and Node.js supports running the webpage the team created.

In the future, the team could optimize the layout of the page, and the running speed of the website. In addition, the team could adjust the colour, text, image and name of the website to improve user experience. More modifications will be added to accommodate the mobile device access. The navigation bar should be set to be touch-screen friendly, and the search and filter buttons should be set to be triggered by signals from the touch-screen.

**11.0 Quality Assurance Description**

Since the team uses waterfall development methodologies, the team continuously integrates the code to keep simplicity, consider users at all times and ensure quality through various testing methods. After the team made new editions on the backend CMS, the team used Insomnia to test all the CRUD methods that were involved in this project. Insomnia could help the team to visualize the semantic errors in a specific CRUD method, so the team could immediately make corrections to have high-quality code in the backend.

Also, the team runs the webpage after connecting the backend and front together to make sure the functional and non-functional requirements are met. The team checked all the icons and data are visible. The webpage overall is organized and easy to navigate through.

Another method to test the web application is that the team could develop different test cases in the CMS to test specific scenarios in a function. If the test cases are running successfully, then the team can conclude the web application quality is assured.

**11.1 Testing Procedures for Project**

* Test using Insomnia

The team uses Insomnia to check how the backend CMS works with CRUD methods. Firstly, the team makes sure the CMS application is running successfully, then the team opens Insomnia to run all the GET, POST, PUT and DELETE requests to make sure the backend functions do not have errors. Here is a list of backend functions below with the testing URLs.

**GetMapping:** Set insomnia to “GET” to read information.

* <http://localhost:8085/player>

This command will return to all the player statistics that are included in the Premier League.

* [http://localhost:8085/player/{name}](http://localhost:8085/player/%7Bname%7D)

This command will return the player information that the player name has been searched for. The name of the player will be put into the {name}.

* [http://localhost:8085/player/search/{playername}](http://localhost:8085/player/search/%7Bname%7D)

This command will return the player information that has been searched. However, the user does not need to type in the exact player’s name. The information could be found by typing a similar word into {name} without case sensitive.

* <http://localhost:8085/team>

This command will return to all the team statistics that are included in the Premier League.

* [http://localhost:8085/team/{name}](http://localhost:8085/team/%7Bname%7D)

This command will return the team information that the team name has been searched for. The name of the team will be put into the {name}.

* [http://localhost:8085/team/search/{teamname}](http://localhost:8085/team/search/%7Bteamname%7D)

This command will return the team information that has been searched. However, the user does not need to type in the exact team name. The information could be found by typing a similar word into {teamname} without being case sensitive.

* <http://localhost:8085/user>

This command will return all the users' information that includes user name, favourite player, and favourite team.

* [http://localhost:8085/user/{username}](http://localhost:8085/user/%7Busername%7D)

This command will return the user information such as user name, favourite player, and favorite team for a specific user. However, the user does not need to type in the exact username. The information could be found by typing a similar word into {username} without case sensitive.

* [http://localhost:8085/userpost/search/team/{teamname}](http://localhost:8085/userpost/search/team/%7Bteamname%7D)

This command will return all the user-posted information that includes a specific team name. Users can type in the team name in {teamname} to get the posts that are related to that specific team.

* [http://localhost:8085/userpost/search/player/{playername](http://localhost:8085/userpost/search/player/%7Bplayername)}

This command will return all the posts about a player according to the player’s name.

The player’s name will be put into {playername}.

* [http://localhost:8085/userpost/search/user/{username}](http://localhost:8085/userpost/search/user/%7Busername%7D)

This command will return all the posts of a user according to the user’s name.

The user’s name will be put into {username}.

**PutMapping:** Set insomnia to “UPDATE”

* [http://localhost:8085/user/{username}](http://localhost:8085/user/%7Busername)

This command will allow users to update their username, user’s password, favorite team and favorite player.

* [http://localhost:8085/userpost/{username}/{Team}/{Player](http://localhost:8085/userpost/%7Busername%7D/%7BTeam%7D/%7BPlayer)}

This command will allow the users to update a specific post that could be found by the username, the team and the player. . The user post's username will be put into {username}, userpost’s team will be put into {Team}, and userpost’s player will be put into {Player}.

**PostMapping:** Set insomnia to “CREATE”

* <http://localhost:8085/user>

This command will create a new user with the information of user’s name, user’s password, user’s favorite team, and the user’s favorite player.The user’s information should be put into insomnia before running this command.

* <http://localhost:8085/userpost>/[{username}](http://localhost:8085/userpost/%7Bid%7D)/{Team}/{Player}

This command will create a new user post with a unique combination of username, team name, and player name as the key. Also, there would be post content included and the whole post would be added to the user post list. The new post’s information should be put into insomnia before running this command.

**Delete:** Set insomnia to “DELETE”

* [http://localhost:8085/userpost/{username}](http://localhost:8085/userpost/%7Bid%7D)/{Team}/{Player}

This command will delete the user’s post according to the id of the user post. The user post's username will be put into {username}, userpost’s team will be put into {Team}, and userpost’s player will be put into {Player}.

* Visualize Web Page by Running Front End

As the team running the websites, we can check how the users use the webpage to meet their needs such as viewing soccer team/player statistics, adding team/player to favourite and read/create/update/delete posts in the forum. The team checked every icon and button to make sure all the data were visible and all buttons were working. The team runs the usability test several times to make sure the website is intuitive to navigate through and it does not make users confused.

* Develop Test Cases for Automated Testing of the Back End

The team has developed three test cases regarding user posts. The test cases include creating new user posts with the related team and/or player names; updating the contents, team and/or player of the user posts; deleting a specific user post in the discussion board.

The team has also developed a test case which checks the read function of the players.

**12.0 Challenges and Difficulties Encountered**

The team has faced several challenges while working on the front-end coding process because the team does not have proficient knowledge of front end development or experience working with front-end before. To solve this problem, the team went over the tutorial notes and searched online to construct the basic structure of front-end code, and add icons and pictures. The team tried to reduce the complexity of the system while keeping all functionalities on the website. Every time the team makes editions such as import data, the internal transition of the home page, or changes of icons and colours, the team goes through the web page to make sure the website is working, otherwise, it will be difficult to find the errors within the codes.

One challenge that needs to be mentioned is that the team was stuck when we were building the login function. The team had issues regarding storing temporary variables about user name information that transfer between different components. The team first tried “$emit” to store temp variables in the front end, however, we found that the “$emit” method works between parent and children parts, but the components in our front are not considered parent and children components. The team considered using the “window” function to create global variables and using the “event bus” to transfer stored information.

**13.0 Lessons Learned**

Overall, the team learned a lot from the experience of building a website about Premier League soccer statistics and forums. The team learned how to utilize the knowledge about RESTful design in the CMS, SQL methods and frontend Vue.js components as the team built up the back-end and front-end of the website. The project provides us with a great opportunity to practice the implementation of CRUD methods and SQL queries to define and manipulate our database. Also, since the team does not have experience building front-end before, the project gave us the opportunity to turn the abstract knowledge we learn into a real website that we can visualize and control. We learned how to build connections between the back-end and front-end, making search bars and login functions, adding icons, and accomplishing the CRUD method at the front-end. Although we had several challenges and difficulties while we were building the website, the team always collaborated together to find the source of errors through testing and discuss the problems together.

Based on our experience, the team found that waterfall development methodology is very helpful for the project to collaborate and communicate with each other. The team learned how the waterfall development methodology principles guide us through the planning, analyzing, designing and implementation stages of the software development life cycle.

In addition, we learned different types of UML diagrams to help us visualize the process, classes, objects and sequence of users’ activities and where the activity takes place. The UML diagrams help the team to organize ideas before and after the coding process. Based on the UML diagrams, the team could further optimize the steps of the tasks in the future.

Thus, the team learned technical skills such as back-end/front-end coding, testing methods, the architecture of the system, the waterfall methodology practices, UML diagrams as well as improved soft skills such as teamwork and communication skills from the website development project.

**Reference:**

[1] *Most popular sport by country 2022. [Online]. Available:*<https://worldpopulationreview.com/country-rankings/most-popular-sport-by-country>. *[Accessed: 19-Oct-2022].*

[2] *Premier League viewing figures*. [Online]. Available:<https://www.sportingindex.com/spread-betting-blog/premier-league-viewing-figures>. [Accessed: 20-Oct-2022].

[3] *Twitter. [Online]. Available:*<https://twitter.com/premierleague>*. [Accessed: 21-Oct-2022].*

[4] “Articles,” *Canadian Soccer News*. [Online]. Available: https://www.canadiansoccernews.com/. [Accessed: 05-Dec-2022].

**Appendix**

**Appendix A: Project Administration Tasks**

| Task | Task Description | Assigned to | Milestone | Due Date | Date Achieved |
| --- | --- | --- | --- | --- | --- |
| 1.0 | Select topic and assign roles | All | Topic | 2022/10/5 | 2022/10/04 |
| 2.0 | Project plan | All | Task identified and assigned | 2022/10/7 | 2022/10/05 |
| 3.0 | Prepare written progress report | All | Project Log updated | 2022/10/21 | 2022/10/14 |
| 3.1 | Introduce context and motivation | Victoria | Group discussion | 2022/10/17 | 2022/10/15 |
| 3.2 | Discussion of requirements | Victoria | Requirements analysis and distribution | 2022/10/17 | 2022/10/15 |
| 3.3 | Discussion of software development methodologies | Richard | Discussion completion | 2022/10/17 | 2022/10/15 |
| 3.4 | Three use cases | All | Use cases completion | 2022/10/12 | 2022/10/11 |
| 3.5 | Context diagram | Jenny | Context diagram completion | 2022/10/12 | 2022/10/11 |
| 3.6 | Level 0 data flow diagram (for 3 use cases) | Rosalind | Diagram completion | 2022/10/15 | 2022/10/13 |
| 3.7 | Level 1 data flow diagram (for 1 use case) | Eric | Diagram completion | 2022/10/15 | 2022/10/13 |
| 3.8 | Discussion of the overall architecture of the system | Ruifan  Eric | Team meeting and discussion | 2022/10/17 | 2022/10/17 |
| 3.9 | project log and discussion of the status implementation | All | Team meeting and discussion | 2022/10/19 | 2022/10/19 |
| 3.10 | Discussion of the challenges, and plans to address them. | All | Team meeting and discussion | 2022/10/20 | 2022/10/20 |
| 4.0 | Prepare project presentation | All | Presentation | 2022/11/30 | 2022/11/30 |
| 5.0 | Assign tasks in written report | All | Report distribution | 2022/11/25 | 2022/11/25 |
| 5.1 | Prepare written project report | All | Report start to be written | 2022/11/26 | 2022/11/26 |
| 5.2 | Overview of web application | All | Completion of the part of the report | 2022/11/26 | 2022/11/26 |
| 5.3 | Summary of requirements | All | Completion of the part of the report | 2022/11/26 | 2022/11/26 |
| 5.4 | All user cases included | All | Completion of the part of the report | 2022/11/26 | 2022/11/26 |
| 5.5 | Identification of design principles | All | Completion of the part of the report | 2022/11/26 | 2022/11/26 |
| 5.6 | Description of current state of web application | All | Completion of the part of the report | 2022/11/26 | 2022/11/26 |
| 5.7 | Description of quality assurance and test procedures | All | Completion of the part of the report | 2022/11/27 | 2022/11/27 |
| 5.8 | Description of any difficulties encountered during the project | All | Completion of the part of the report | 2022/11/27 | 2022/11/27 |
| 5.9 | Lessons learned | All | Completion of the part of the report | 2022/11/28 | 2022/11/28 |
| 5.10 | Reference | All | Completion of the part of the report | 2022/11/28 | 2022/11/28 |
| 5.11 | Appendices | All | Completion of the part of the report | 2022/12/04 | 2022/12/04 |

Note:

For all the tasks and subtasks, the team will first do a discussion on the content with all the members, after we reach a consensus on the content, the member assigned with the tasks/subtasks will finish the tasks/subtasks.

The team will keep updating and add tasks/subtasks to this list as we progress on the project, therefore some of the blank areas are left as TBD, e.g., subtasks in task 5.0.

Section B: Web Application Development Task List

| Task  ID | Task | Task Assigned To | Milestone Due Date | Date Achieved |
| --- | --- | --- | --- | --- |
| 1.0 | Create database | All | 2022/10/06 | 2022/10/06 |
| 1.1 | Research of initial datasets | Victoria  Richard | 2022/10/07 | 2022/10/07 |
| 1.2 | Compiling database | Eric  Rosalind | 2022/10/15 | 2022/10/15 |
| 1.3 | Establish relations and declare primary/foreign keys | Jenny  Ruifan | 2022/10/20 | 2022/10/20 |
| 2.0 | Discussion and planning of overall architecture. | All | 2022/10/14 | 2022/10/14 |
| 2.1 | Plan of SDLC methodologies | All | 2022/10/14 | 2022/10/14 |
| 2.2 | Identify and detail project requirements | All | 2022/10/16 | 2022/10/16 |
| 2.3 | Create use cases | All | 2022/10/17 | 2022/10/17 |
| 3.0 | Constructing back-end architecture | All | 2022/10/20 | 2022/10/20 |
| 3.1 | Construct the Content management system | All | 2022/10/23 | 2022/10/23 |
| 4.0 | Constructing front-end architecture | All | 2022/10/26 | 2022/10/24 |
| 4.1 | Establish connection between back-end and front-end | All | 2022/10/27 | 2022/10/25 |
| 5.0 | UI/UX design of the web application | All | 2022/11/25 | 2022/11/23 |
| 6.0 | Create test cases and record test results | All | 2022/12/04 | 2022/12/04 |

Note:

The team will keep updating and add tasks/subtasks to this list as we progress on the project, since we are still in progress of learning the software development cycle, there are uncertainties for what tasks we must accomplish in order to establish a complete web application; more subtasks will come as we finish the task 2.0.

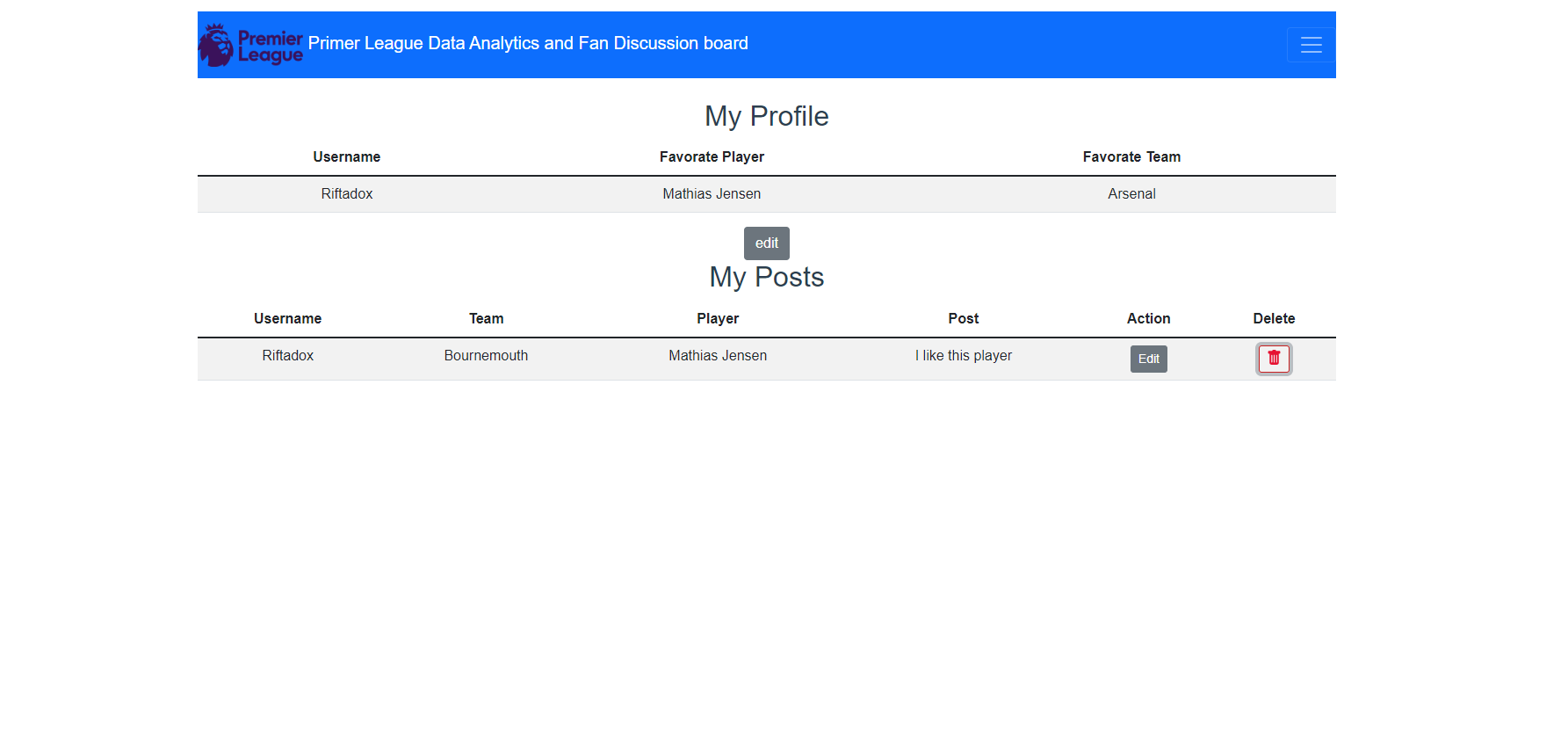
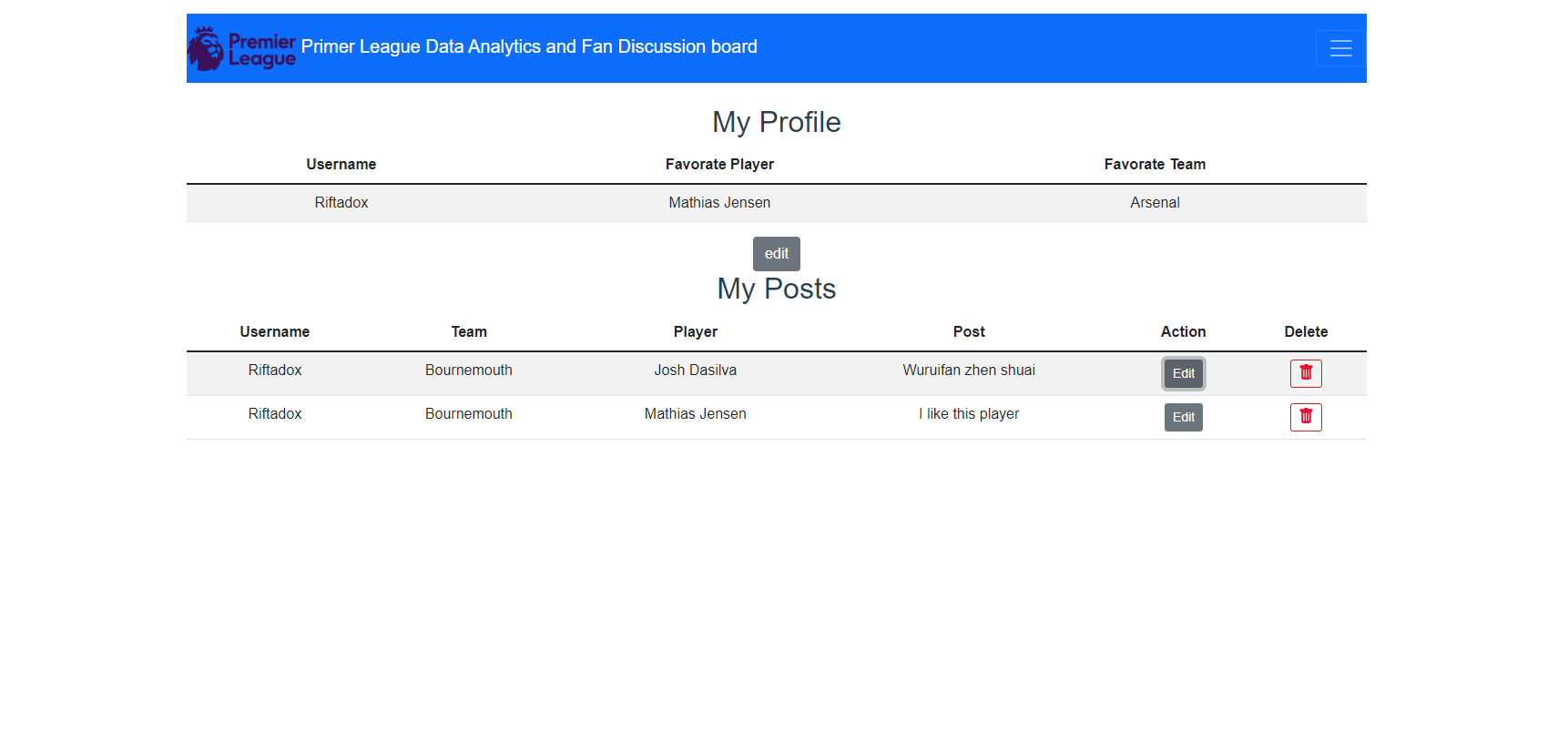
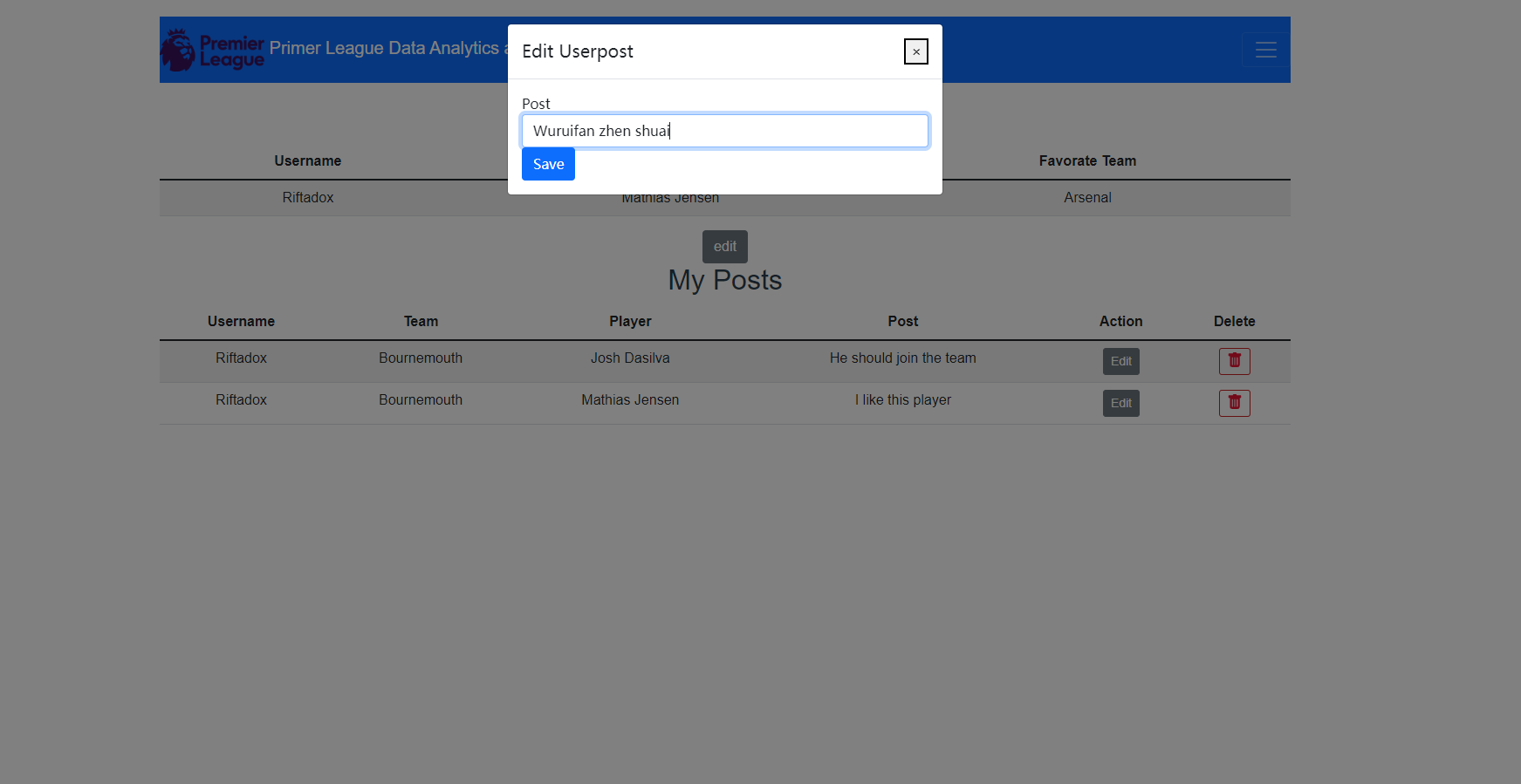
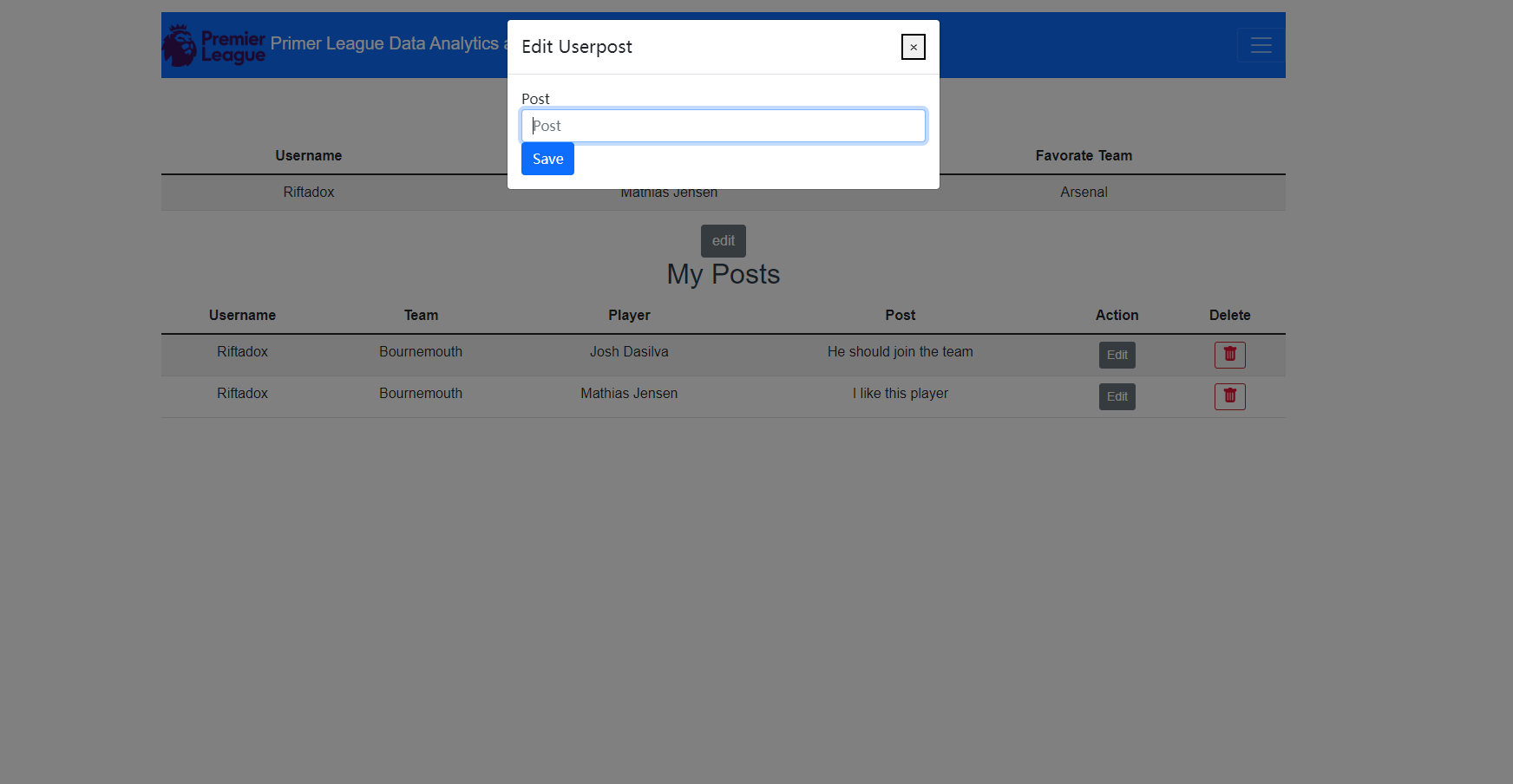
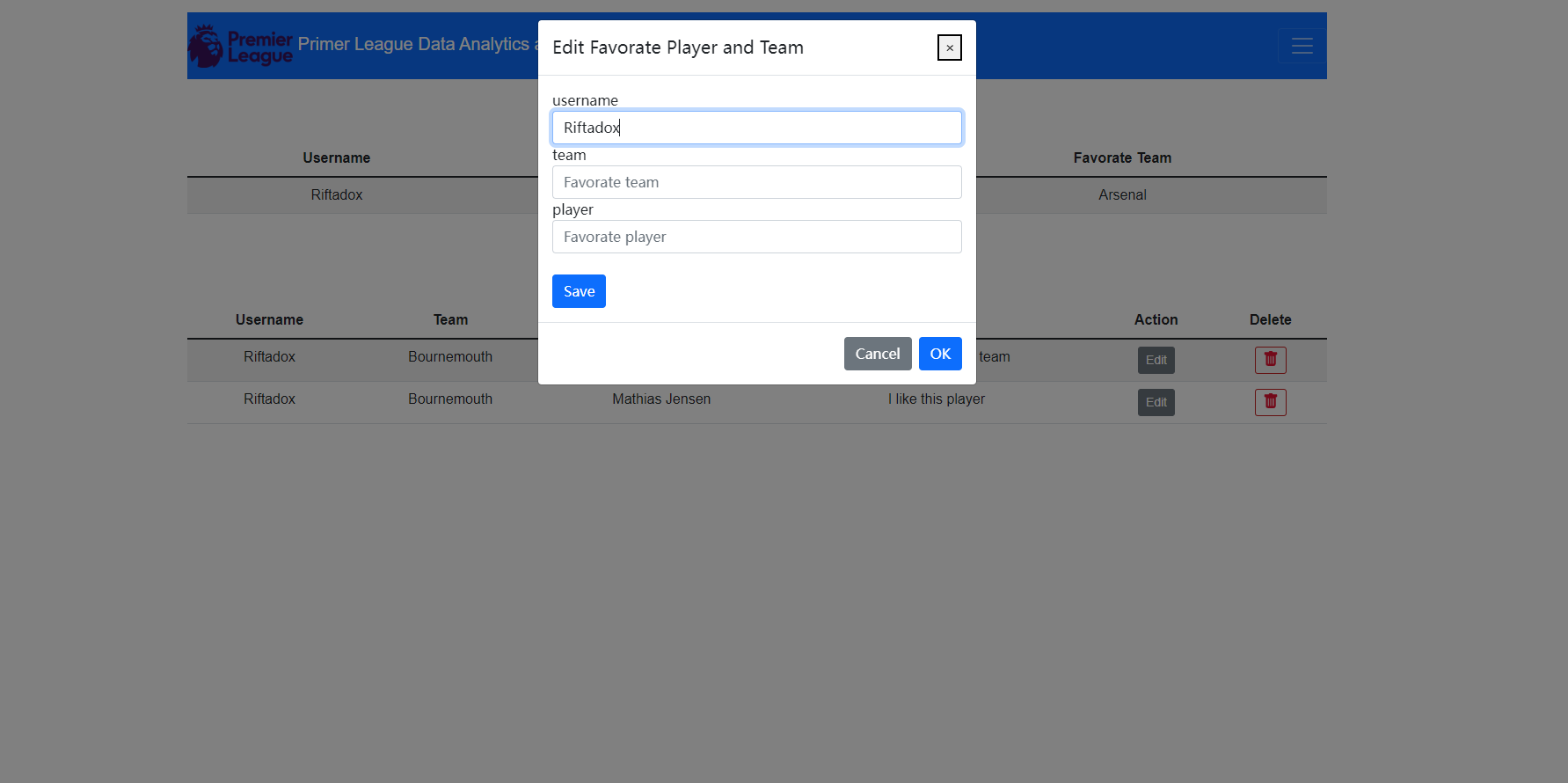
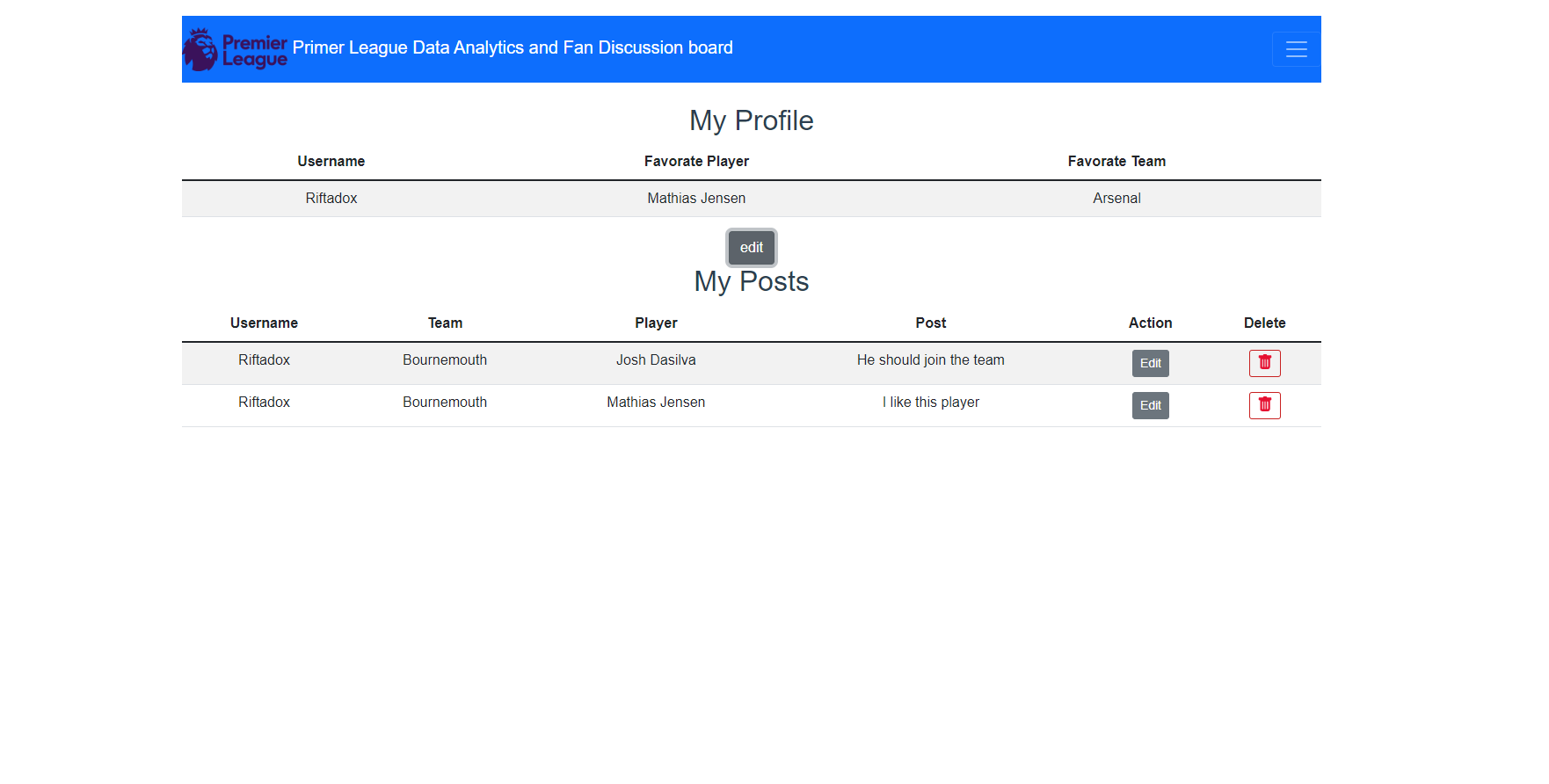
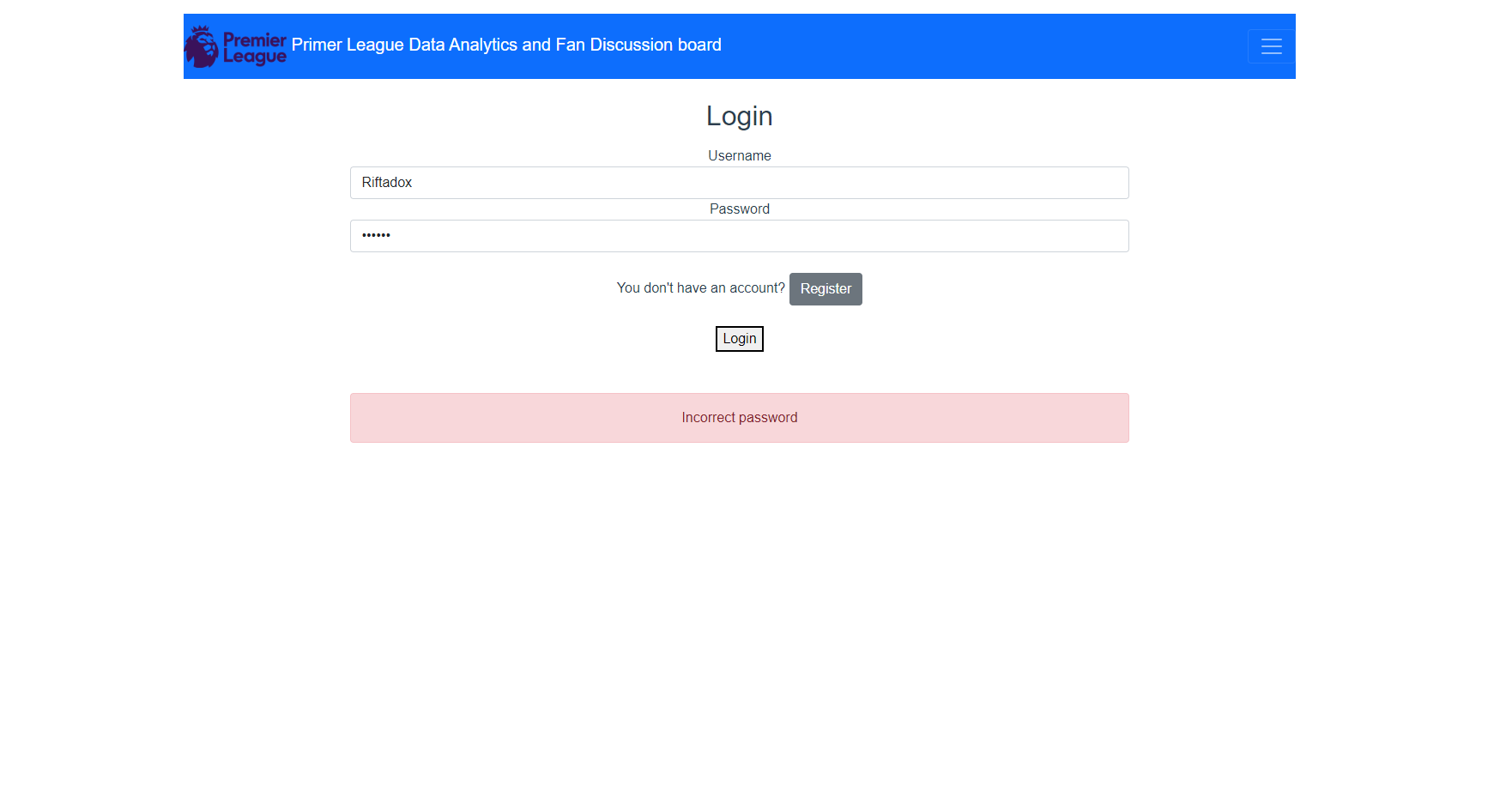
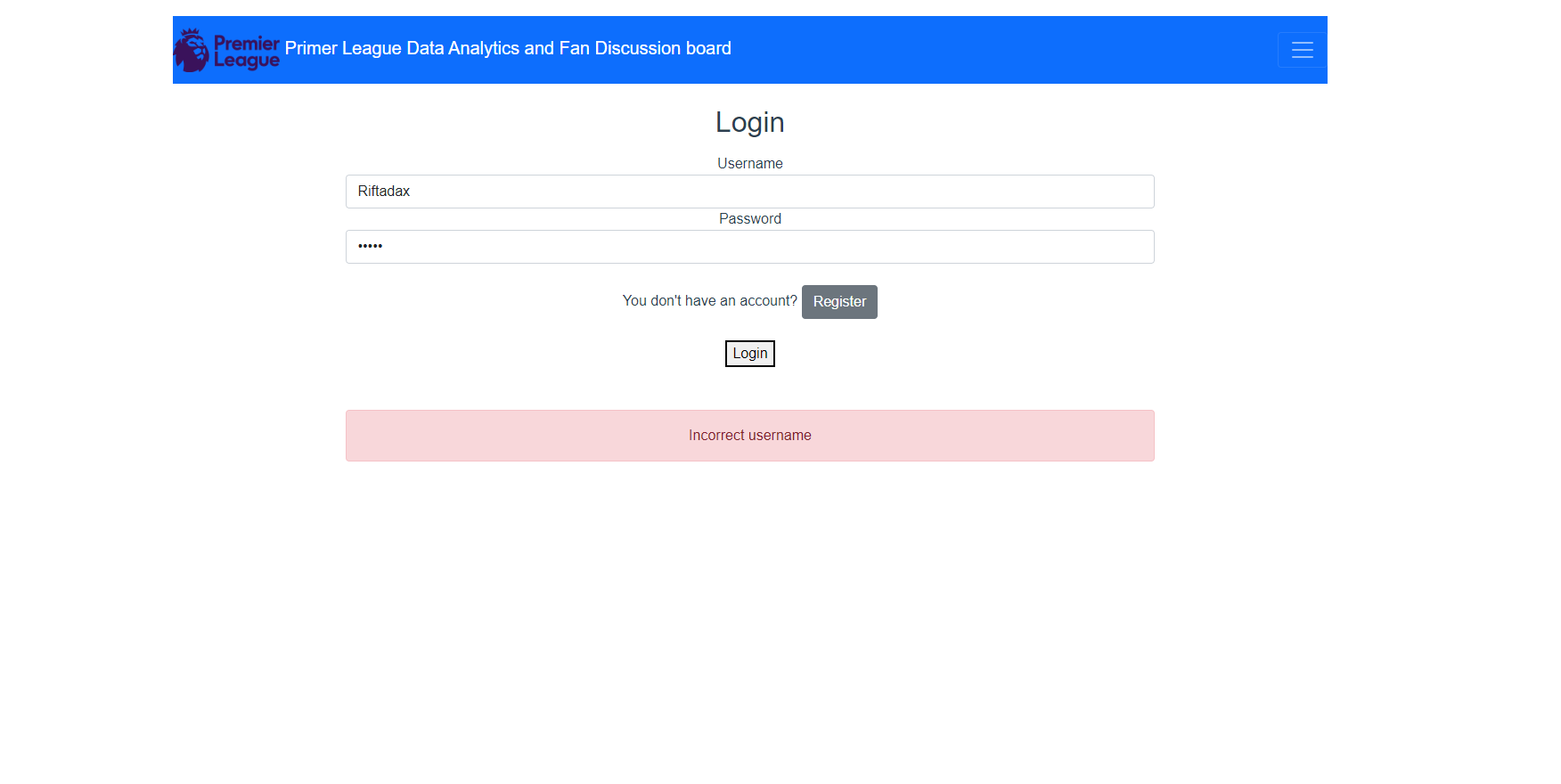
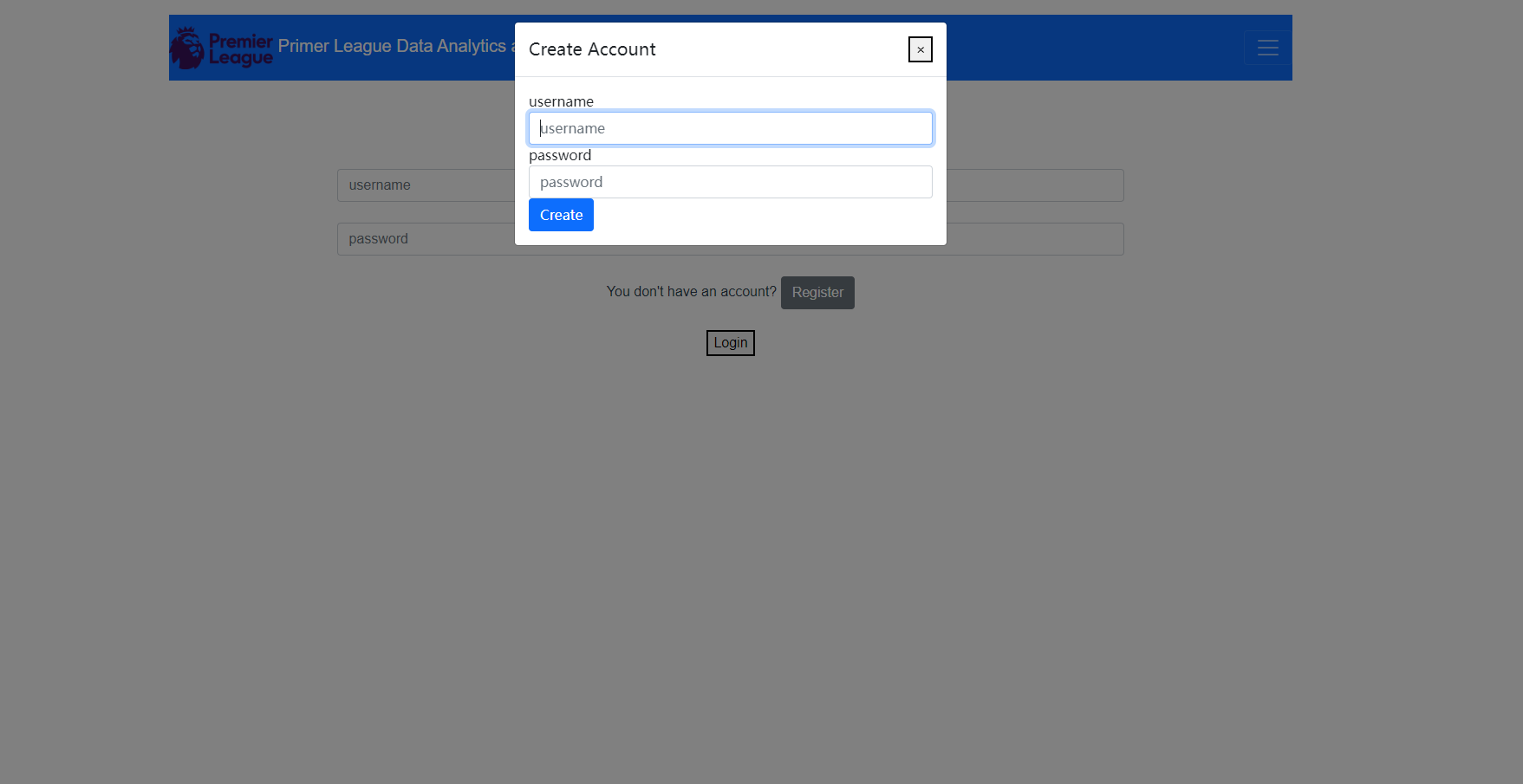
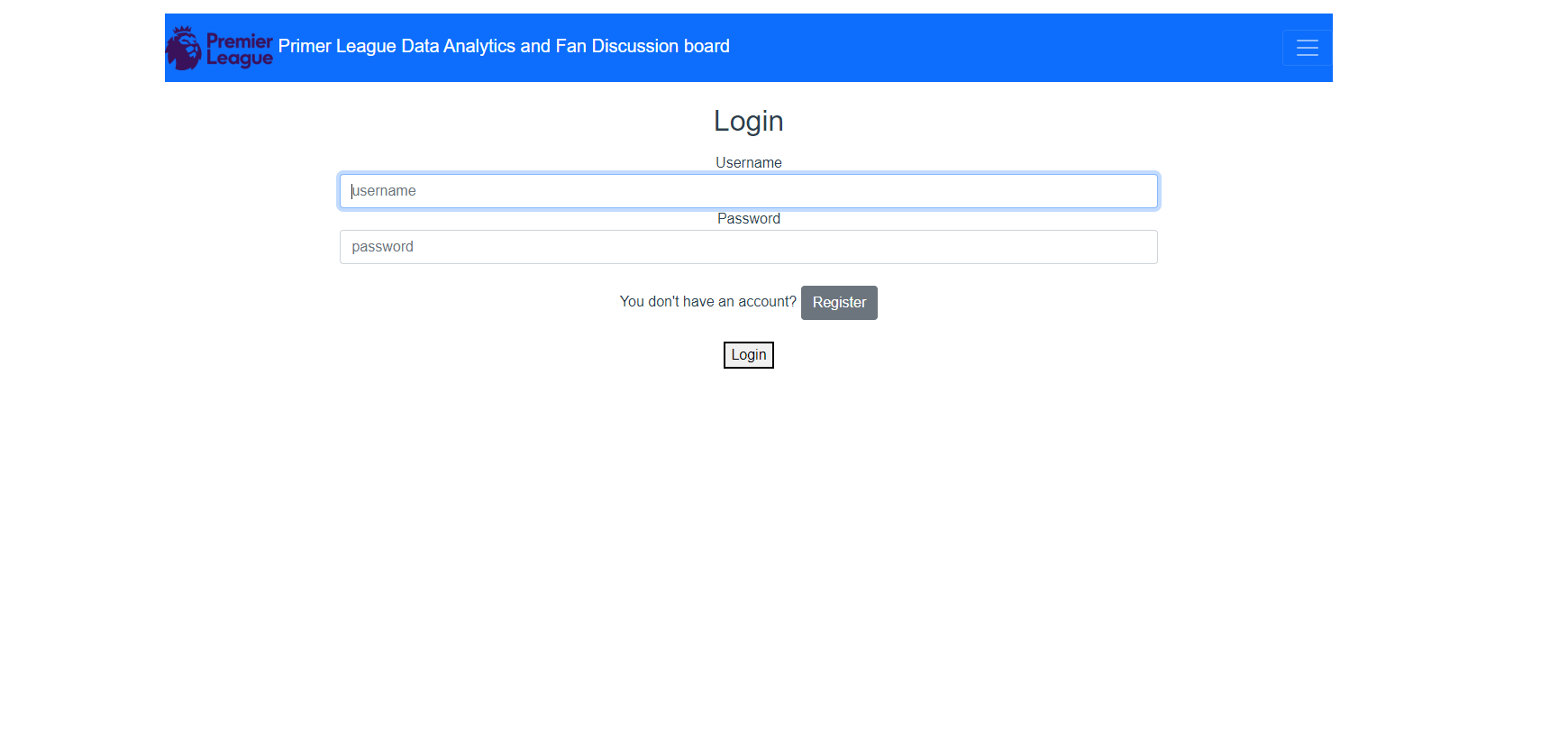
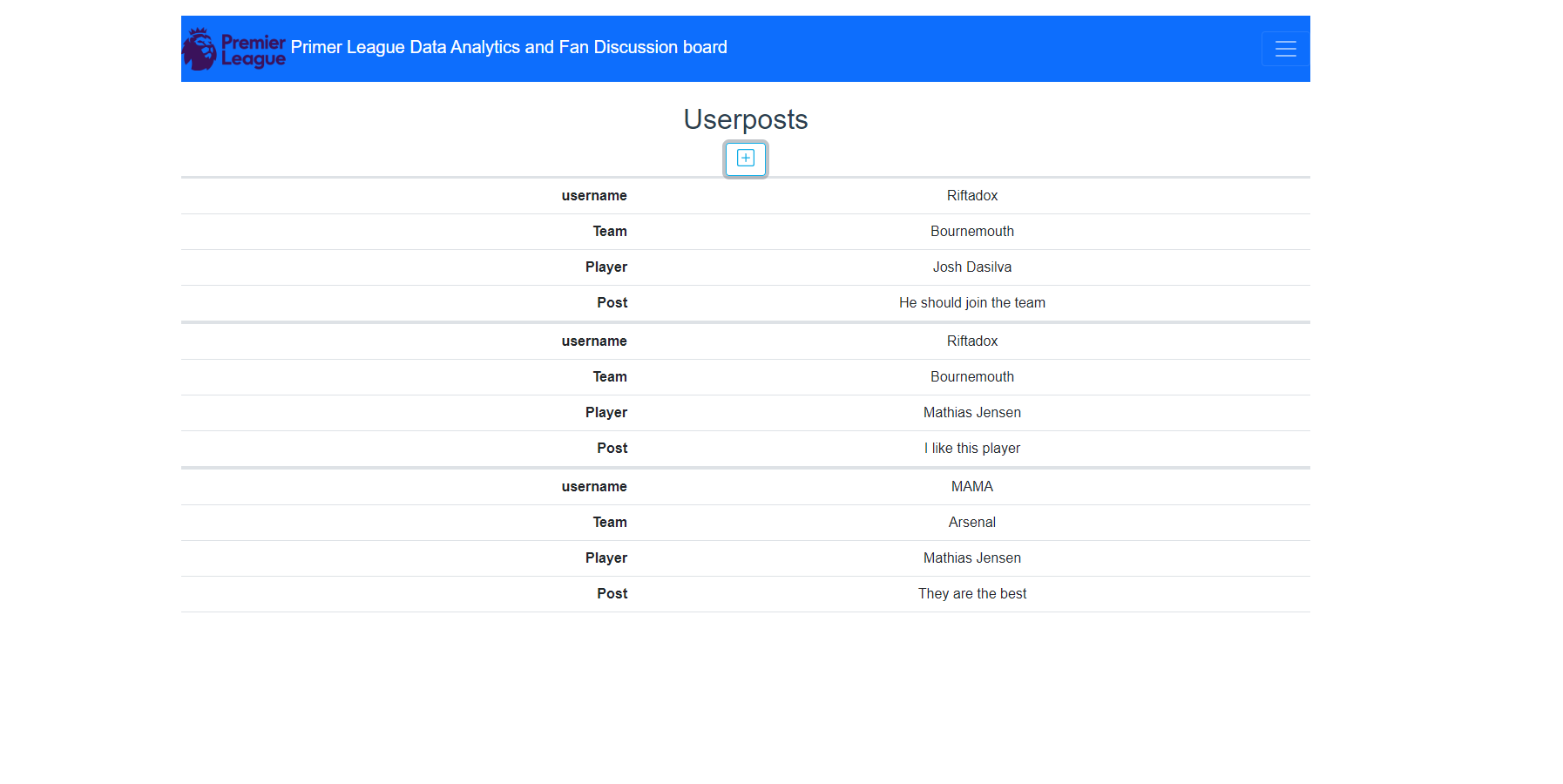
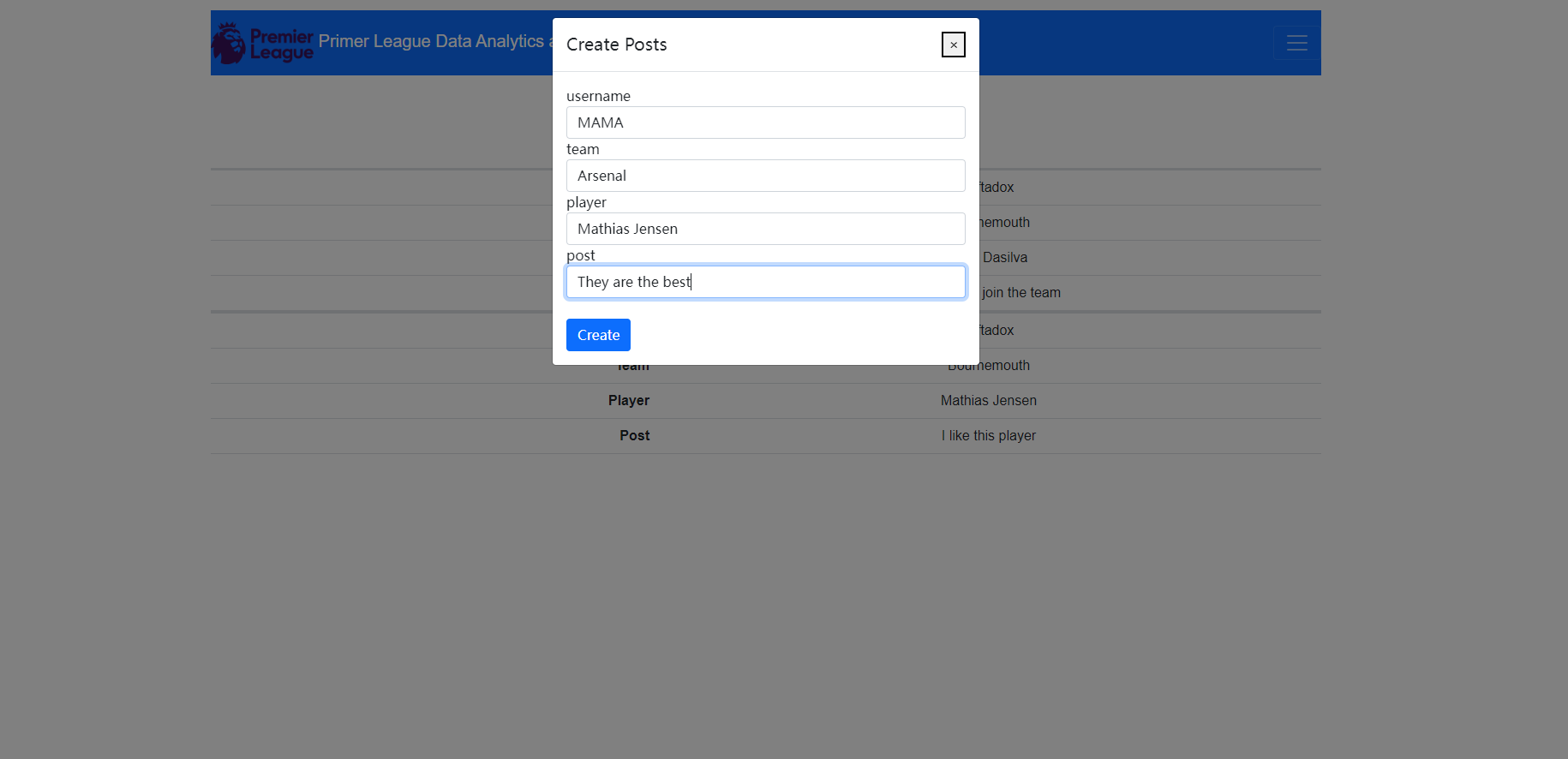
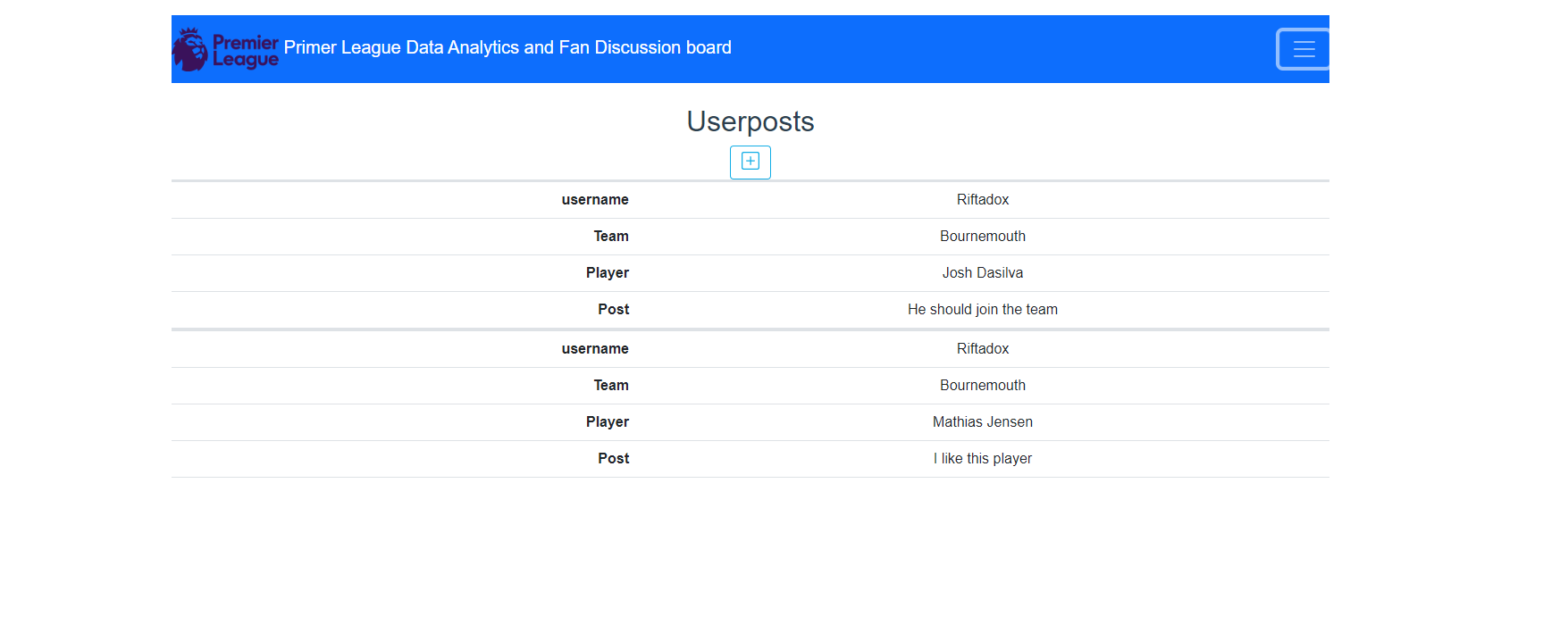
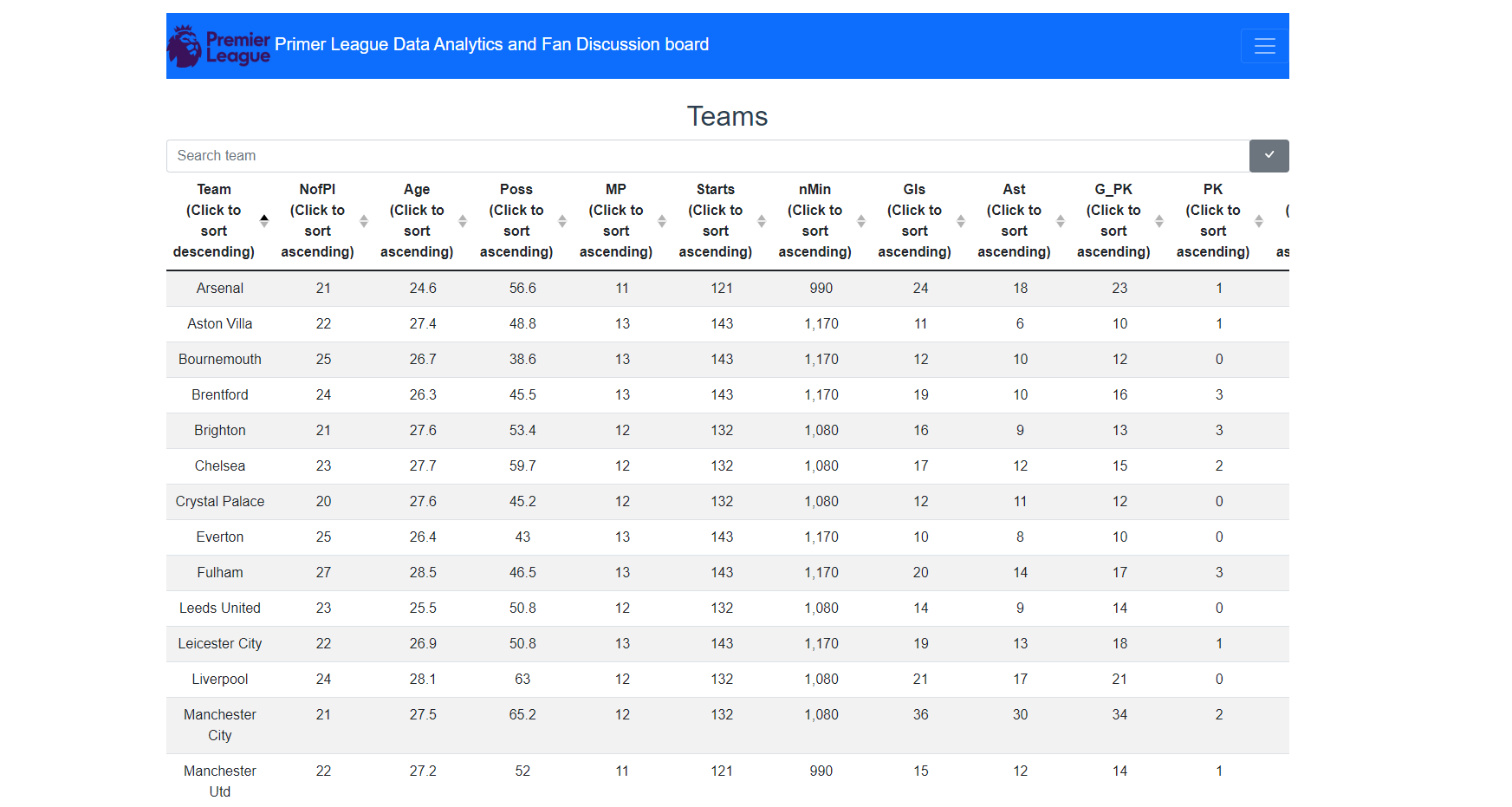
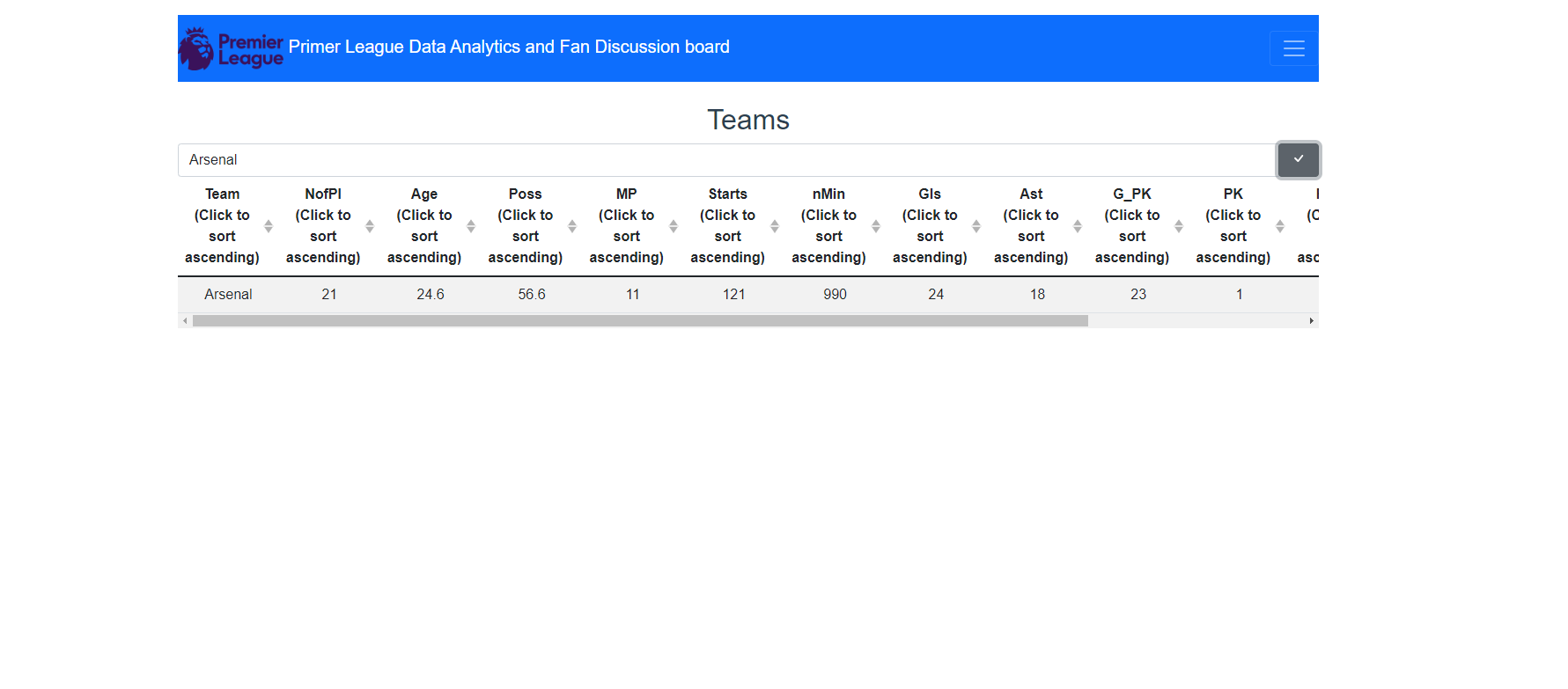
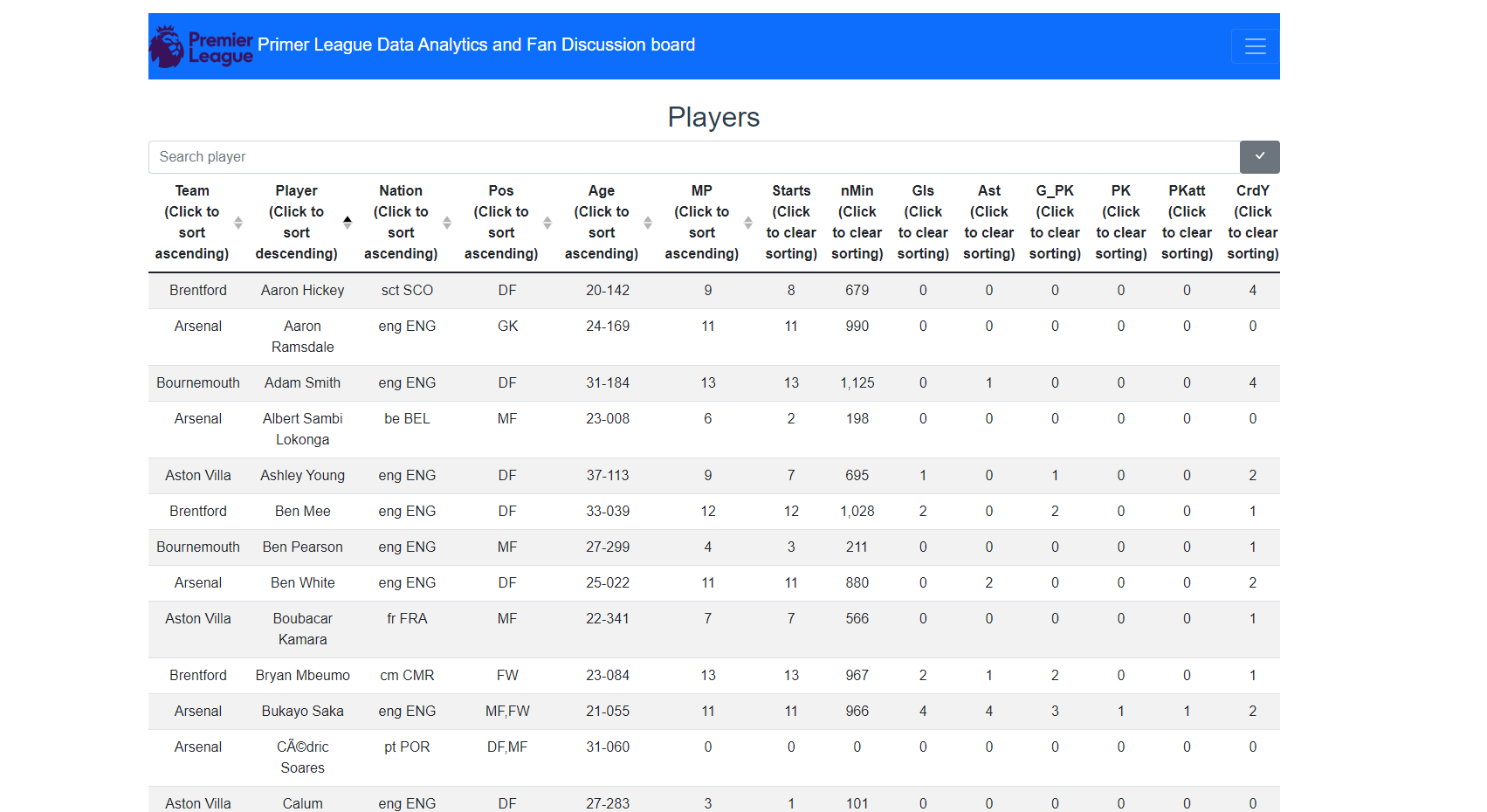
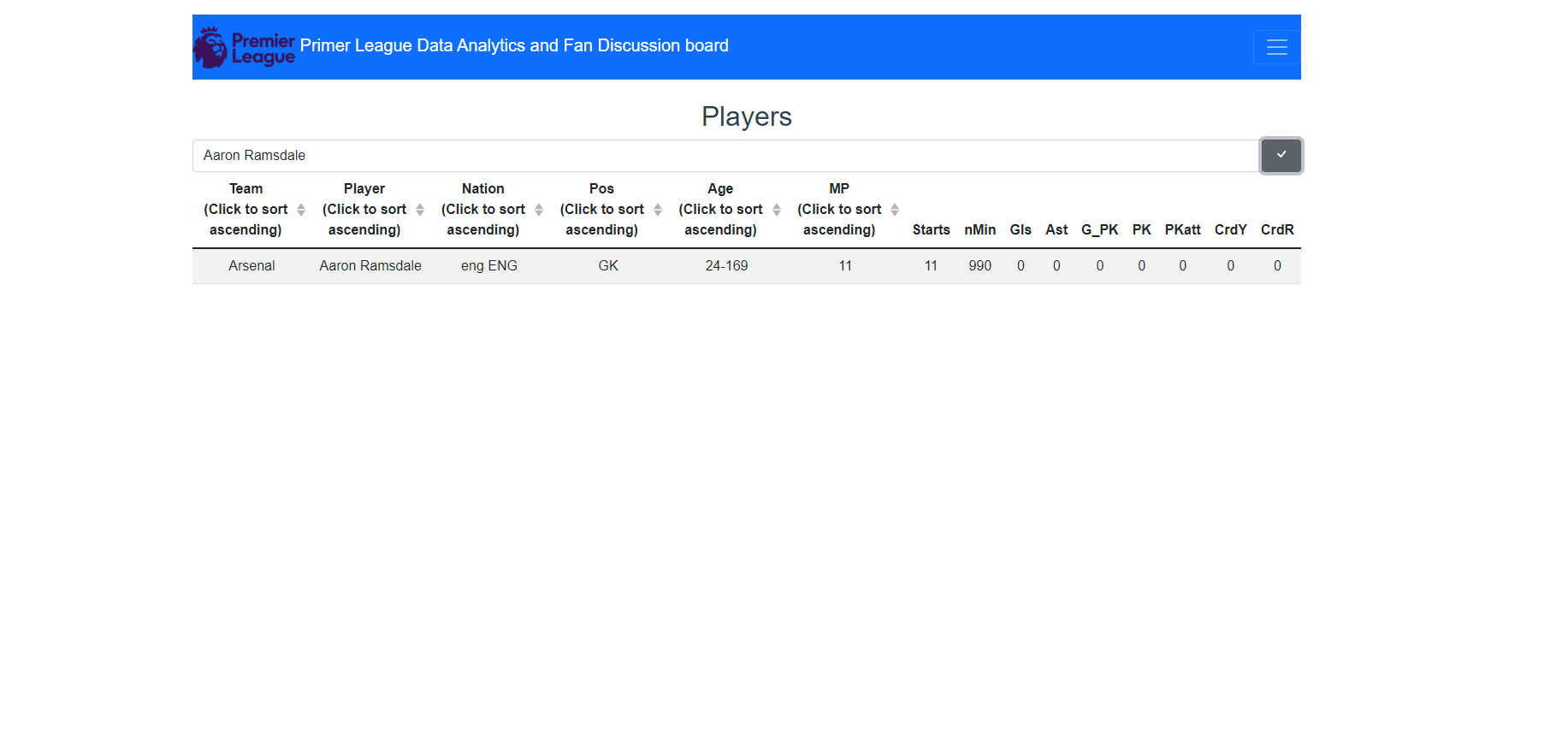
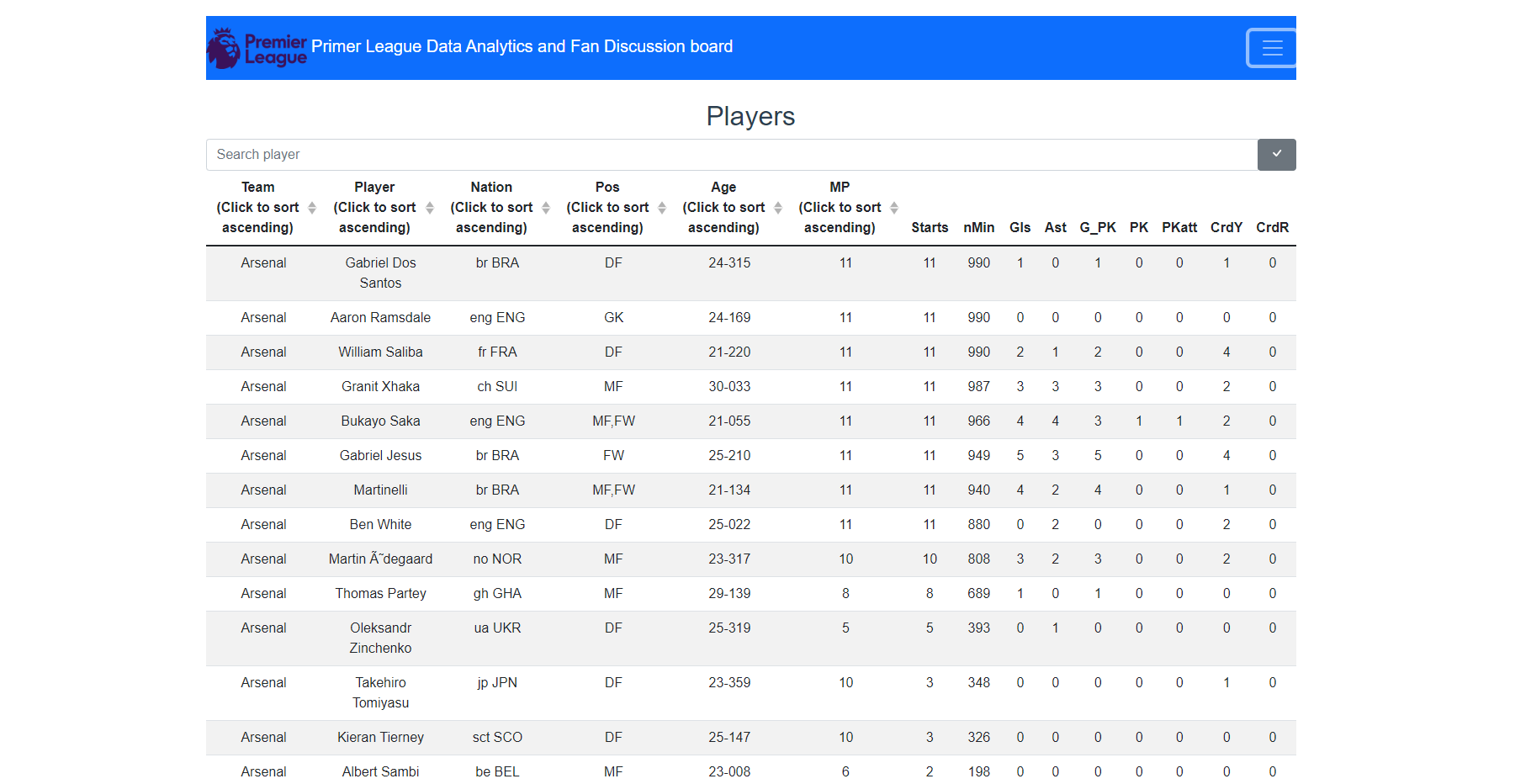
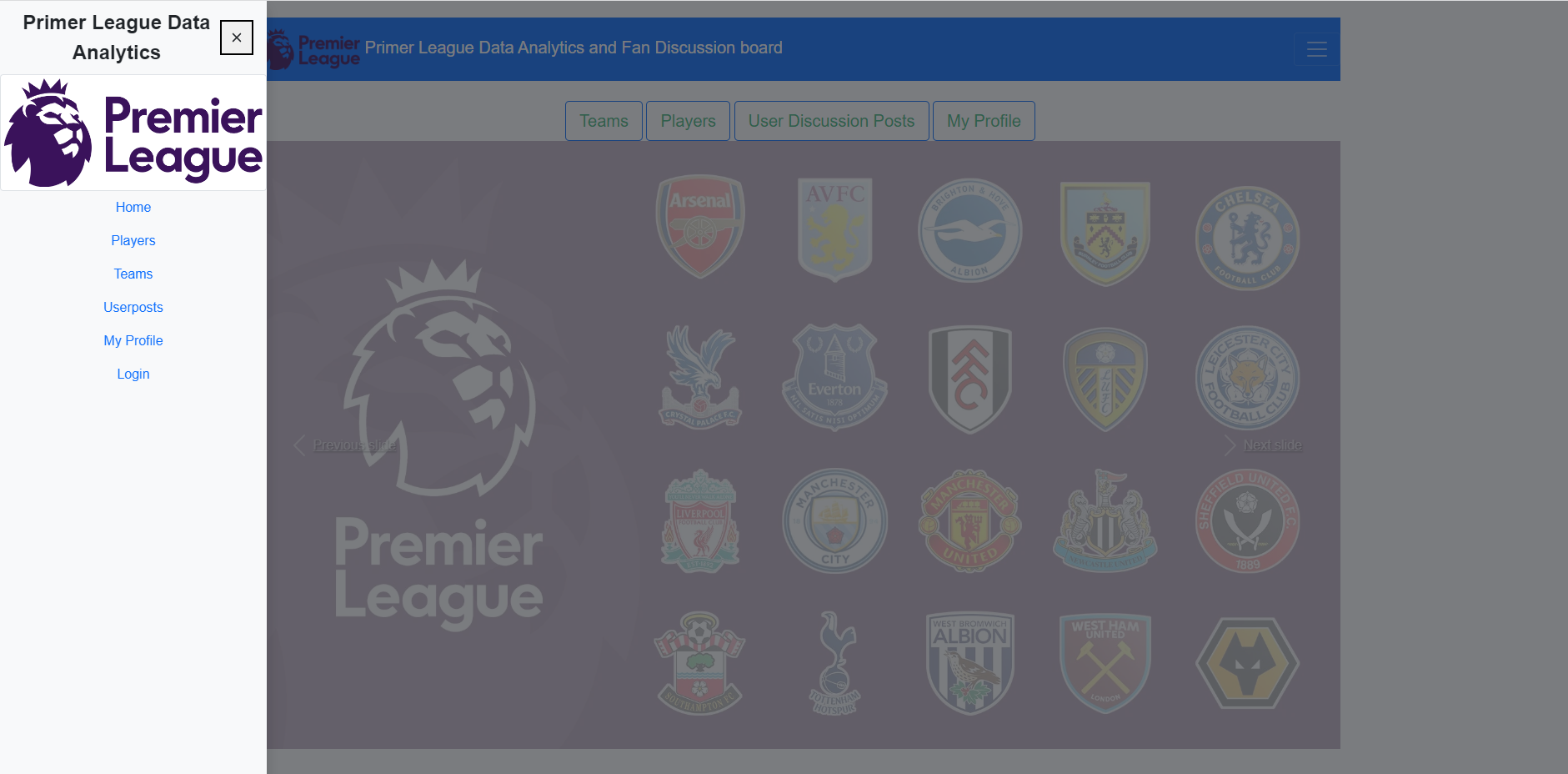
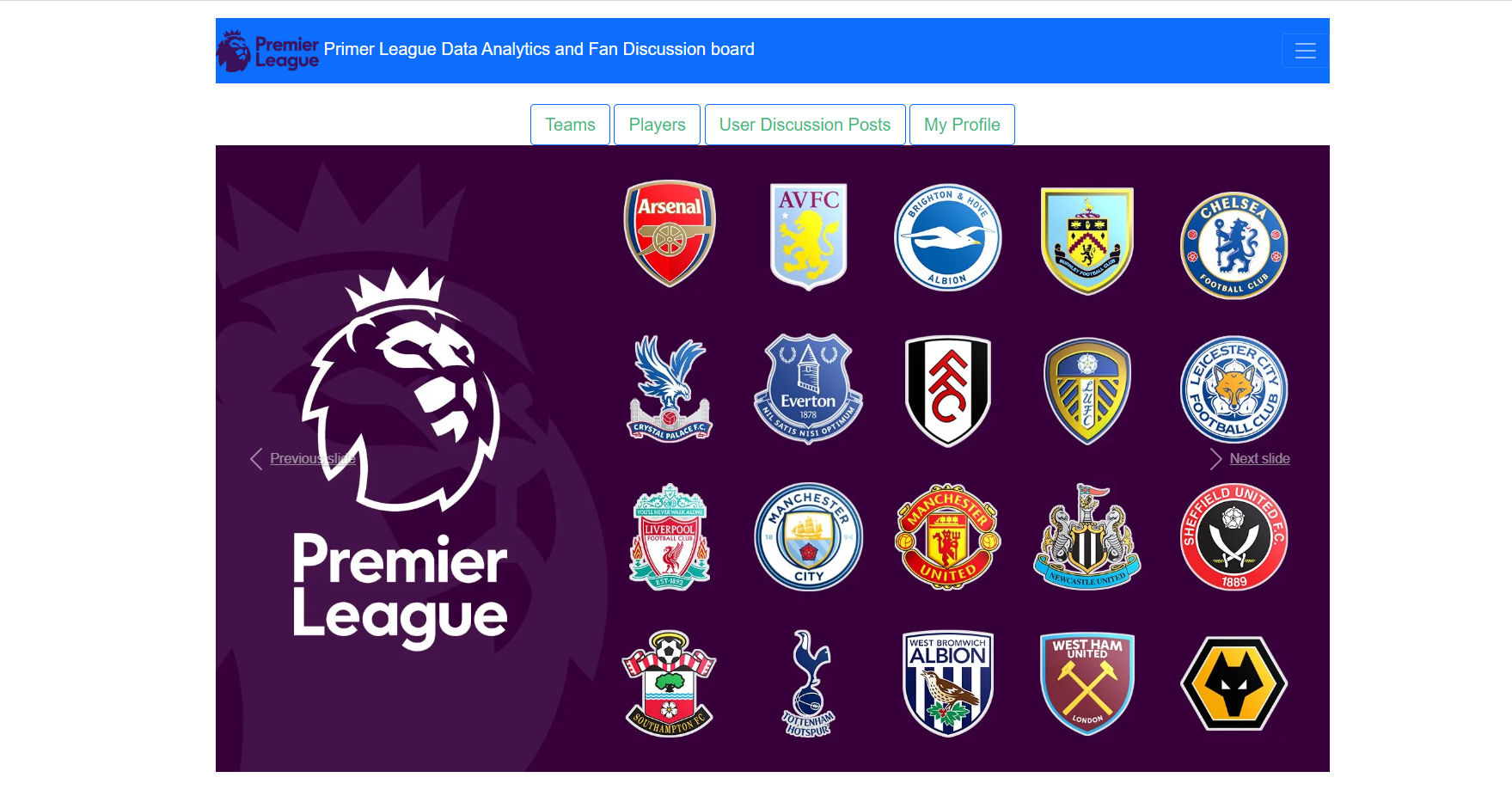
Section C: Progress Summary

| Task | Date | Status | Comments |
| --- | --- | --- | --- |
| A1.0 | 2022/10/5 | Completed! | The topic is selected to build an online community site for Primer League （Soccer） data analysis |
| A2.0 | 2022/10/7 | Completed! | Project administration tasks are expanded and detailed; Web app development tasks are identified, and tasks are assigned to team members. |
| A3.0 | 2022/10/20 | Completed! | The team had finished the preliminary search and identified target users and functions, and used three use cases to demonstrate the Lfunctionality. Three data flow diagrams were then created to visualize the architecture of the system. |
| A4.0 | 2022/11/30 | Completed! | Each group member constructed their own slides and written scripts. Rehearsal has been completed. |
| A5.0 | 2022/11/25 | Completed! | Report is divided into sections and assigned to each group member. |
| B1.0 | 2022/10/10 | Completed! | The databases had been created to store player and user information. |
| B2.0 | 2020/10/14 | Completed! | The blueprint of the project is constructed with duties assigned to each group member. |
| B3.0 | 2022/10/20 | Completed! | The back-end is well developed with all CRUD functions being tested through insomnia. |
| B4.0 | 2022/10/25 | Completed! | Front-end and back-end connection was established and the web application could be accessed from the browser. |
| B5.0 | 2022/11/23 | Completed! | Frontend UI/UX design has been finalized. |
| B6.0 | 2022/12/04 | Completed! | Automated tests were created to test CRUD functionalities of the application. |

Note:

This is the progress of the project and will be updated bi-weekly as we process the project.

**Appendix B: Copy of the Website**



**Appendix C: Detailed description of the roles and responsibilities of each team member**

| **Name** | **Role** | **Reponsibility** |
| --- | --- | --- |
| Ricky Wu | Back-end Lead | Design the overall architecture of the back end. Ensure that back-end related functions can perform well before each deadline. |
| Jenny You | Front-end Lead | Design the overall architecture of the front end. Ensure that front-end-related functions can be performed well before each deadline. |
| Richard Jing | Back-end engineer | Perform back-end knowledge and help the team to construct a back-end that fulfills the functions that the team’s website needs to have. |
| Victoria Li | Research | Research the necessary information that the team needs in developing the team’s website. Also, research information for writing the final report. |
| Rosalind Wang | UI/UX designer | Design the UI for the team’s website. Make sure that the UI displayed to our users should be user-friendly and could be well-understood. |
| Eric Wang | Front-end engineer | Perform front-end knowledge and help the team to construct a front-end that fulfills the functions that the team’s website needs to have. |
| Muchen Liu | Project manager | Submit files, and push on others for deadlines. |

**Appendix D: Copy of the Presentation Material**