BILKENT UNIVERSITY

SPRING 2017 - CS 353

TERM PROJECT FINAL REPORT

Football Database Management System

GROUP 13

21301854 Ömer Akgül
21302483 Ali Osman Çetin
21300994 Ali Günes
21401058 Orhun Kar

cs353.github.io

1-) A description of your application system

The football database management system is an online application used to manage data related to the sport of football. The system especially focuses on the transfer system of football. We represent the parties involved in football transfers with users types in our program. A normal user type also exists to represent anyone that is not involved actively but observes the transfers as they happen.

There are 4 unique user types portrayed in the system. These are normal user, coach, club manager, player agent. The normal user is allowed to monitor all public game data. This includes but not limited to viewing past transfers, viewing current teams of the players, viewing match data, league statuses.

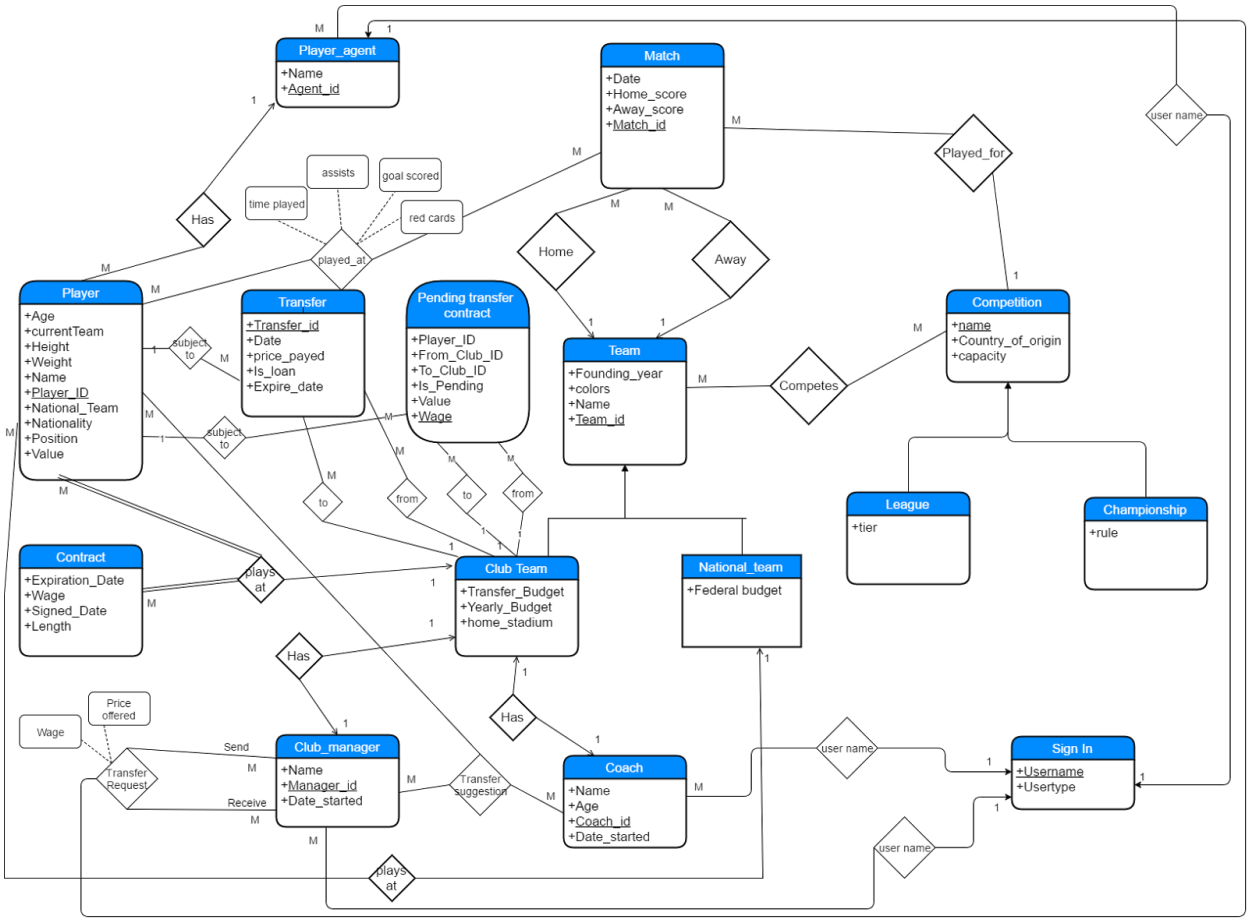
The coach users are very similar to a normal user except they can recommend transfers to be done to the club manager of their teams. This request feature exists only to inform the club manager and does not include a role in the transfer process itself if the club manager decides to ignore it.

The club managers are one of the two parties that realize a transfer. The club managers are able to request transfers to teams and approve or decline transfer requests coming in. The club manager also have the ability to see if the coach wants a transfer. Club managers also have the ability to edit the values of the players bound to their teams.

The player agent is the other party that makes a transfer happen. Agents can accept or decline an offer coming in for the one of the players represented by them. They are able to see all coming in transfer requests.

All of the the users use the same application interface however, the pages they view change according to the user type as seen in the user manual section.

2-) Final E/R


Player_agent
+Name
+Agent_id
Match
+Date
+Home_score
+Away_score
+Match_id
user name
Played_for
Has
assists
goal scored
time played
red cards
played_at
Home
Away
Player
+Age
+currentTeam
+Height
+Weight
+Name
+Player_ID
+National_Team
+Nationality
+Position
+Value
Transfer
+Transfer_id
+Date
+price_payed
+Is_loan
+Expire_date
Pending transfer contract
+Player_ID
+From_Club_ID
+To_Club_ID
+Is_Pending
+Value
+Wage
Team
+Founding_year
+colors
+Name
+Team_id
Competes
Competition
+name
+Country_of_origin
+capacity
subject to
to
from
League
+tier
Championship
+rule
Contract
+Expiration_Date
+Wage
+Signed_Date
+Length
plays at
Club Team
+Transfer_Budget
+Yearly_Budget
+home_stadium
National_team
+Federal budget
Has
Price offered
Wage
Send
Receive
Transfer Request
Club_manager
+Name
+Manager_id
+Date_started
Transfer suggestion
Coach
+Name
+Age
+Coach_id
+Date_started
user name
Sign In
+Username
+Usertype

3-) Final list of tables

3.1-) Player

Relational Model

player (player_id, age, height, weight, name, national_team_id, club_team_id, agent_id,

nationality, position, value)

Functional Dependencies:

national_team_id -> nationality

Candidate Keys:

player_id

Foreign Keys:

agent_id, natioal_team_id, club_team_id

Normal Form:

Boyce-Codd Normal Form

3.2-) Player Agent

Relational Model

player_agent(agent_id, name)

Functional Dependencies:

-

Candidate Keys:

agent_id

Foreign Keys:

Normal Form:

Boyce-Codd Normal Form

3.3-) Transfer

Relational Model

transfer(transfer_id, player_id, from_team_id, to_team_id, price_payed, date, is_loan,

expire_date)

Functional Dependencies:

If is_loan then expire_date exists.

Candidate Keys:

trasnfer_id

Foreign Keys:

player_id, from_team_id, to_team_id

Normal Form:

Boyce-Codd Normal Form

3.4-) Contract

Relational Model

contract(signed_date, player_id, team_id, expiration_date, wage)

Functional Dependencies:

-

Candidate Keys:

(signed_date, player_id, team_id)

Foreign Keys:

player_id, team_id

Normal Form:

Boyce-Codd Normal Form

3.5-) Club Team

Relational Model

club_team(club_team_id, name, colors, coach_id, manager_id, home_stadium, founding_year,

transfer_budget, yearly_budget)

Functional Dependencies:

-

Candidate Keys:

club_team_id

Foreign Keys:

coach_id, manager_id

Normal Form:

Boyce-Codd Normal Form

3.6-) National Team

Relational Model

national_team(national_team_id, name, colors, founding_year, federal_budget)

Functional Dependencies:

name->

Candidate Keys:

national_team_id

Foreign Keys:

-

Normal Form:

Boyce-Codd Normal Form

3.7-) Club Manager

Relational Model

club_manager(manager_id, name, date_started)

Functional Dependencies:

-

Candidate Keys:

manager_id

Foreign Keys:

Normal Form:

Boyce-Codd Normal Form

3.8-) Coach

Relational Model

coach(name,age,coach_id, team_id date_started)

Functional Dependencies:

-

Candidate Keys:

coach-id

Foreign Keys:

team_id

Normal Form:

Boyce-Codd Normal Form

3.9-) League

Relational Model

league(name, capacity, country_of_origin, tier)

Functional Dependencies:

-

Candidate Keys:

-

Foreign Keys:

-

Normal Form:

Boyce-Codd Normal Form

3.10-) Championship

Relational Model

championship(name, capacity, country_of_origin, rule)

Functional Dependencies:

-

Candidate Keys:

-

Foreign Keys:

-

Normal Form:

Boyce-Codd Normal Form

3.11-) Match

Relational Model

match(match_id, home_score, away_score, home_team_id, away_team_id, date,

competition_name)

Functional Dependencies:

-

Candidate Keys:

match_id

Foreign Keys:

home_team_id, away_team_id, competition_name

Normal Form:

Boyce-Codd Normal Form

3.12-) Player Played at Match

Relational Model

player_match(player_id, match_id, time_played, goal_scored, assist, red_cards)

Functional Dependencies:

-

Candidate Keys:

{(player_id, match_id)}

Foreign Keys:

Player_id, match_id

Normal Form:

Boyce-Codd Normal Form

3.13-) Club League

Relational Model

club_league(club_id, league_name)

Functional Dependencies:

-

Candidate Keys:

{(club_id, league_name)}

Foreign Keys:

club_id, league_name

Normal Form:

Boyce-Codd Normal Form

3.14-) Coach to Club Manager Suggest

Relational Model

transfer_suggest(player_id, club_id)

Functional Dependencies:

-

Candidate Keys:

{(player_id, club_id)}

Foreign Keys:

Player_id, club_id

Normal Form:

Boyce-Codd Normal Form

3.15-) Pending Transfer-Contract Request

Relational Model

pending_transfer_contract(player_id, from_club_id, to_club_id, is_pending, value, wage)

Functional Dependencies:

-

Candidate Keys:

{(player_id, from_club_id, to_club_id)}

Foreign Keys:

player_id, from_club_id, to_club_id

Normal Form:

Boyce-Codd Normal Form

4-) Implementation Details

The reported football transfer management system is a web application system. Currently, the system is not online however it is accessible through our github repository[1].

The web interface was implemented using the Bootstrap front-end framework and coded in HTML and CSS. Bootstrap allowed us to design functional but simple user interfaces for each user.

For the back-end needs of the web application we decided to use the back-end python framework Django. This framework allows us to create and update models easily and implement functionality into the web pages using python. This decision was made in order to make the back-end development process convenient while also keeping functionality.

We use SQlite as our database management system. We chose SQlite because the application -being a course project- will not undergo any substantial traffic. Being a lightweight application we do not need a separate process for the database management like MySql or oracle provides. SQlite is ideal for us since it is a fully functional database management system but does not have much overhead.

SQlite is connected to Django in the background so by using python we are able to interact with the database as well as the the user interface.

[1] https://github.com/omer58/ftms

5-) Advanced Database Features

5.1-) Conflict Check (Trigger)

- Each time when a coach assigned to a team a trigger will check if there is already a coach for that team.
- A player will only have one player agent so a trigger will control that when an agent assigned to a player
- A team coach will suggest a player to Club Manager only once. A trigger will check that and will drop the request from the manager's suggestion list.

5.2-) Information Update (Trigger)

- After each match relevant attributes of a Player will be updated. Such as goal_per_game, avg_time_played etc.
- League standings will be updated according to results after each match.

5.3-) Contract Cancellation (Trigger)

- A trigger will be constructed for players so that it will check whether contract is expired or not by comparing the *current_date* and *Expiration_date*

5.4-) Finding the Total Values of Leagues (Report)

- Using the 'value' attribute of player, a new table will be constructed by summing all teams player's value in a league. After that a new table will be constructed as LeagueValues to access those values.

5.5-) Finding players who scored X goals, assists etc. in a league (Report)

- A report will be constructed to show players for example who scored 10 goals 5 assists or more in a league. Also separate reports will be constructed by using goals and assists etc. This may help coaches to suggest new players to the team.

5.6-) Using secondary indices for Player names (Secondary indices)

- Since searches for players will be done by using the names of those players instead of their ids which is primary key, we conclude that this implementation will increase the overall performance of our project by reducing the access time to players

5.7-) Calculating Total Team Points in Leagues (Stored Procedures)

- In this database standard procedure will be used for total point calculation. *Home_score* and *Away_score* variables will be collected with unique *match_id.* Each teams points will be calculated with comparing the goal amounts. If the difference is greater than 0, 3 point will be added, if it is 0, 1 point will be added. These procedures will applied to all teams. So this will be a stored procedure.

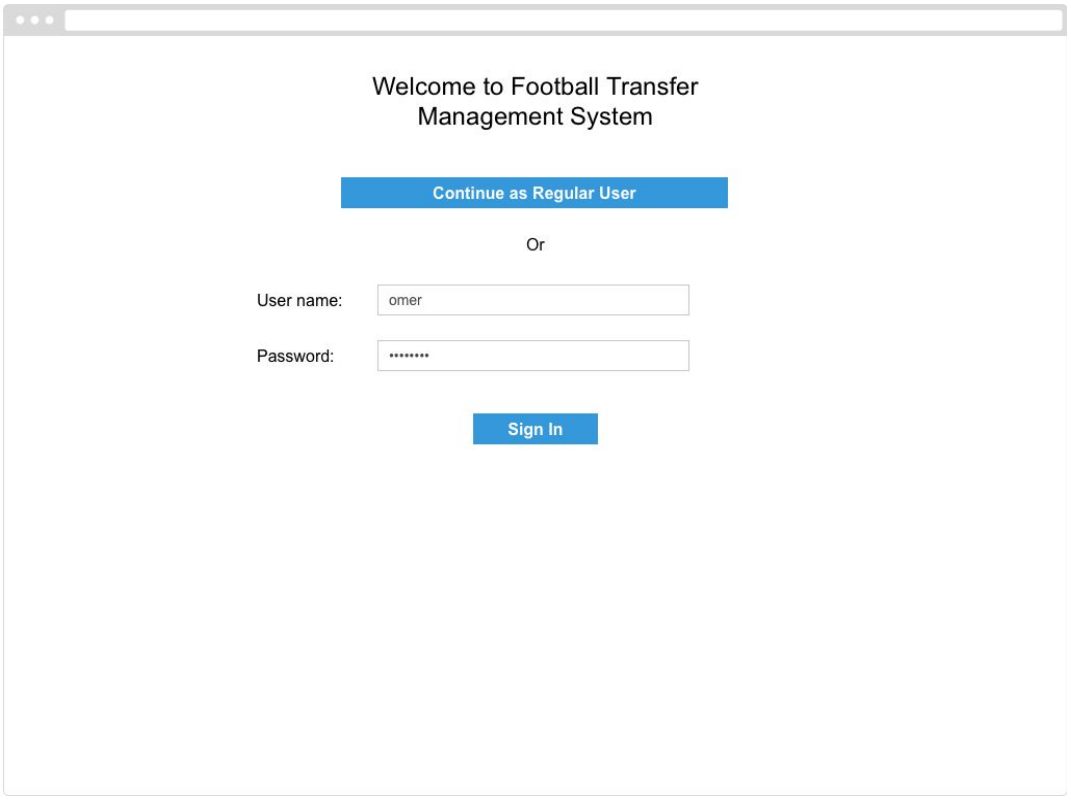
6-) User’s Manual(describe the usage of the system)

There are four types of users who have different specific features in the system. These types of users are normal users, club coaches, player agents and club managers. Other three types have every features of the normal user, and additional features according to the user type. The one cannot register to the system, this means a person can only sign in with his/her pre-provided username and password.

6.1-) Normal User

6.1.1-) Sign In

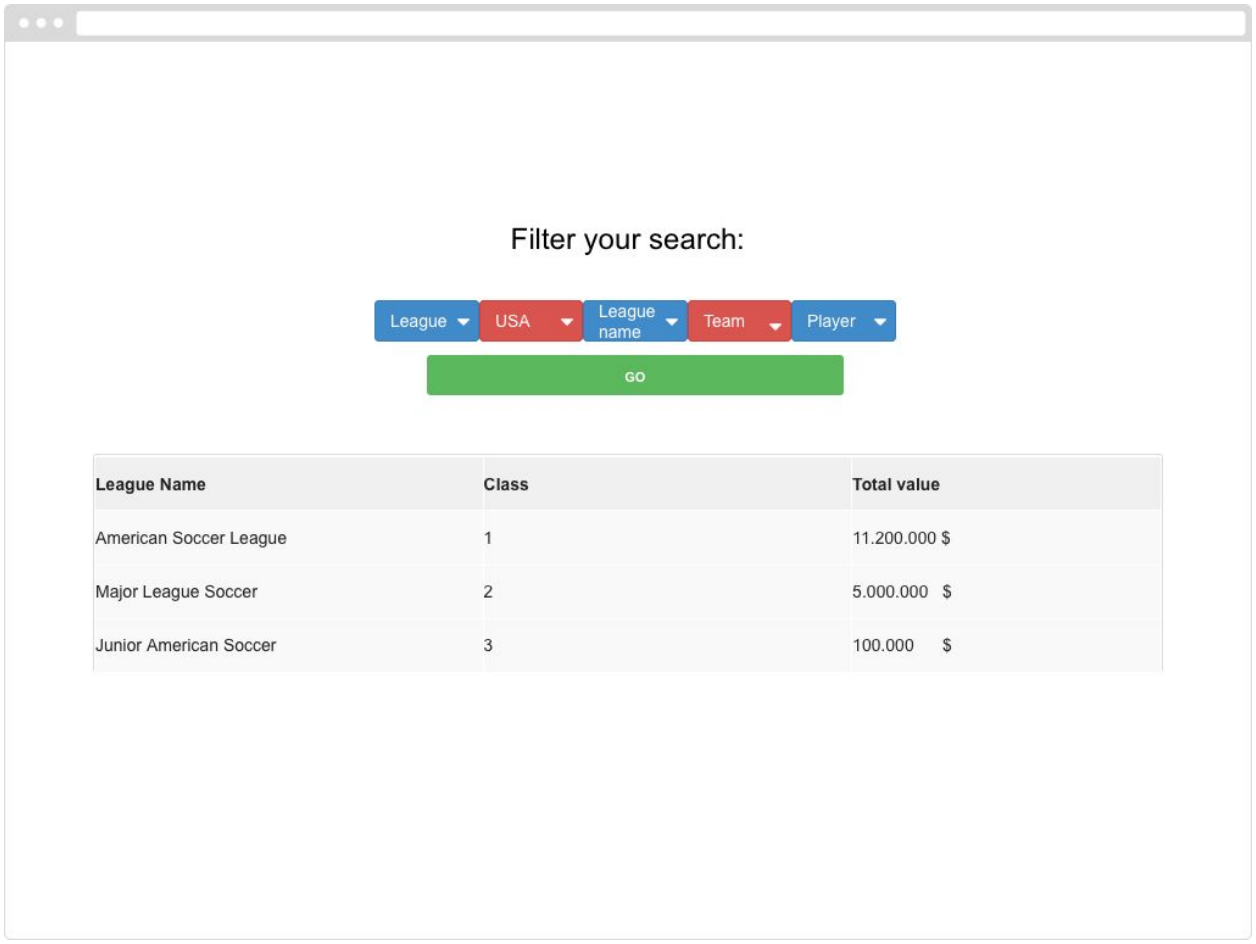
Normal User is a user type who do not need to sign in, this means you can reach information about leagues, teams and players without signing in. When you open the website, first page you encountered is this welcome page:



If you are a regular user, you need to click “Continue as Regular User” and you can reach to the other pages.

6.1.2-) Home Page

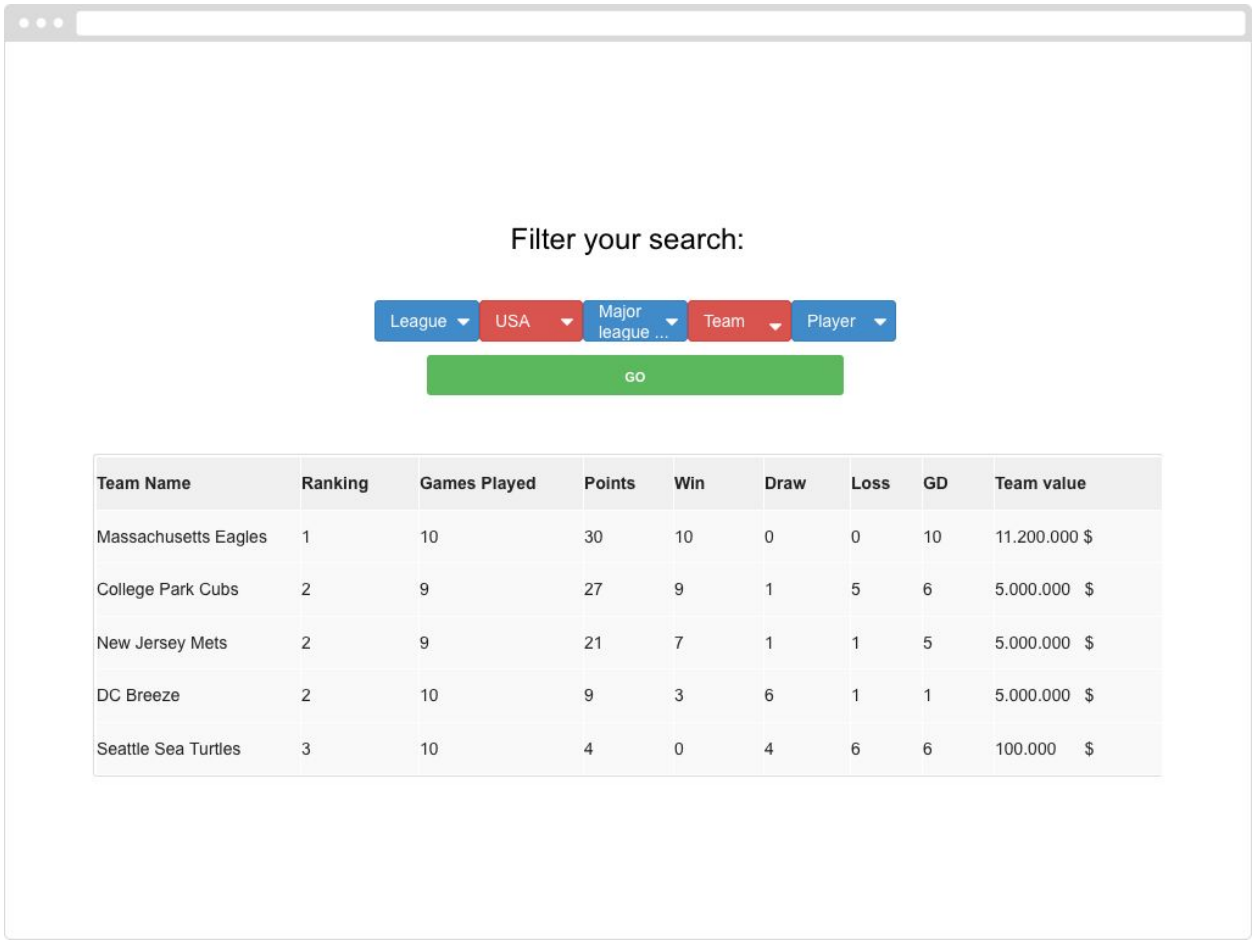
When you click “Continue as Regular User” button, the first page you encountered is list of the soccer leagues or championships, and some information about these leagues existing in the system. You can filter your search by selecting league or championship and the nation of the league without changing of the page.



If you filter the search by selecting the league name, you will continue with the new page.

6.1.3-) League Standings and Match Details

In this page, there are rankings of the teams in selected league. The league table which can be seen in this page includes games played, points, wins, draws and losses of every team. Also, you can see the total value of any team in this league.

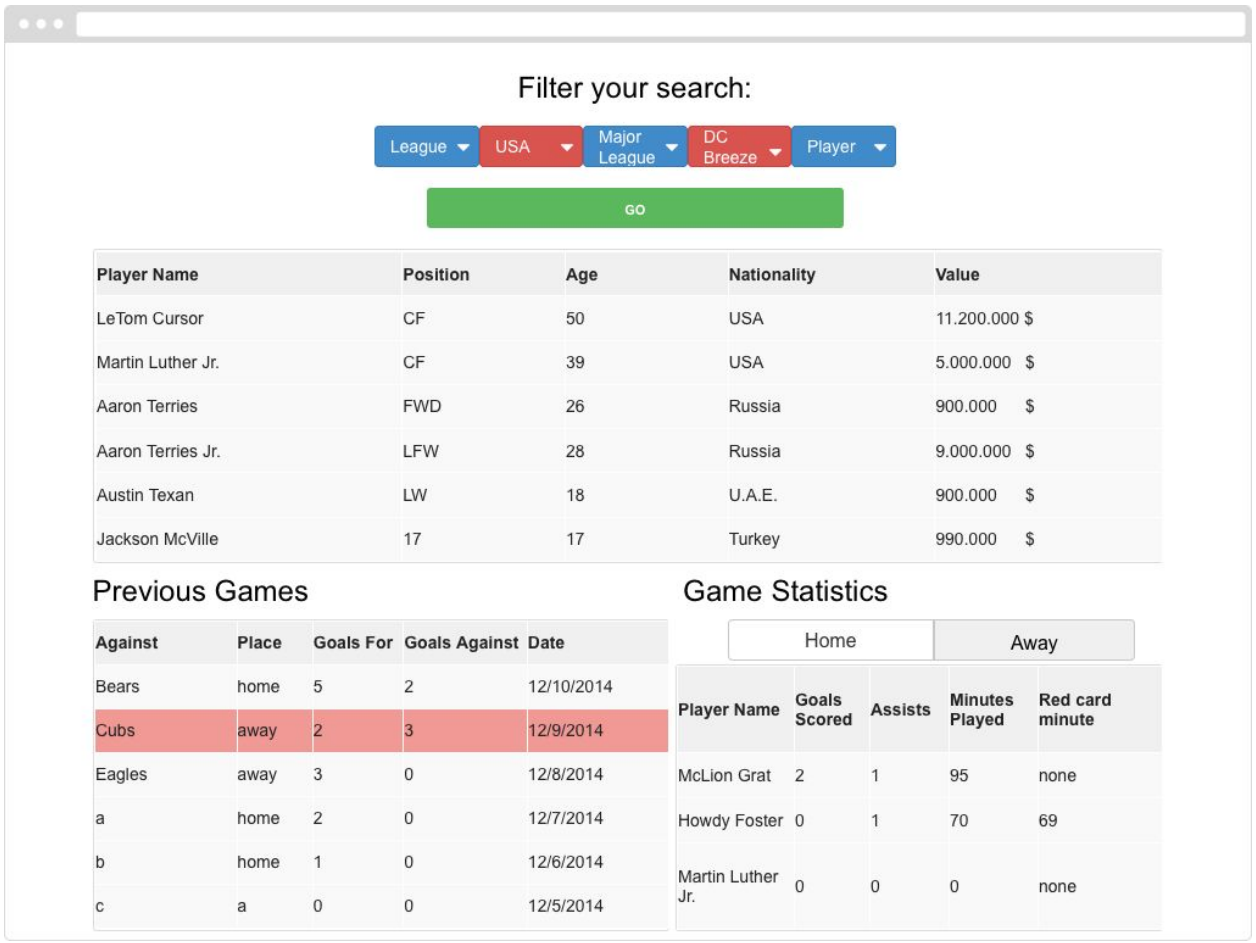


Team Name	Ranking	Games Played	Points	Win	Draw	Loss	GD	Team value
Massachusetts Eagles	1	10	30	10	0	0	10	11.200.000 $
College Park Cubs	2	9	27	9	1	5	6	5.000.000 $
New Jersey Mets	2	9	21	7	1	1	5	5.000.000 $
DC Breeze	2	10	9	3	6	1	1	5.000.000 $
Seattle Sea Turtles	3	10	4	0	4	6	6	100.000 $

You can filter the search by selecting one of the team in this league, and continue with the new page which contains the team information.

6.1.4-) Team Details, Past Matches and Statistics

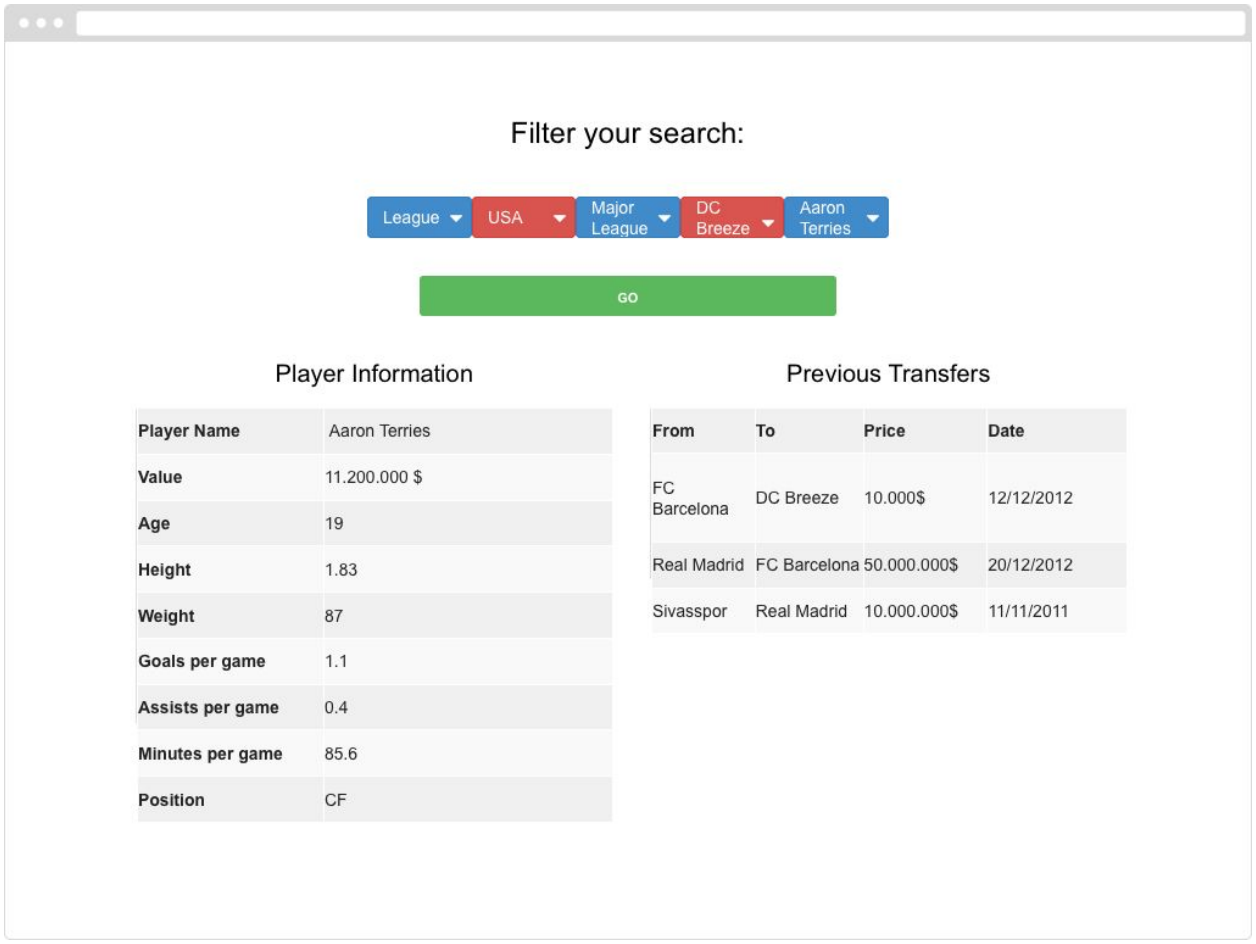
In this page, there is a table of all players of the selected team. Not only the name list of the players, but also some additional information which are position, age, nationality and value of each player is shown in this table. Played games information of this team is shown in another table on the same page. You can select a game to see the match statistics of this game, which are included goals scored, assists, minutes played and red card information of every player in another table as well.



You can filter the search by selecting a player in this team, and and continue with the new page which contains detailed information about this player.

6.1.5-) Player Details and Previous Transfers

In this page, there are two tables about player information and previous transfer information of that player. Player information table includes not only general information, but also player statistics, which are goals per game, assists per game and minutes played per game. Previous transfers table includes the team transferred from, the team transferred to, transfer price and transfer date.

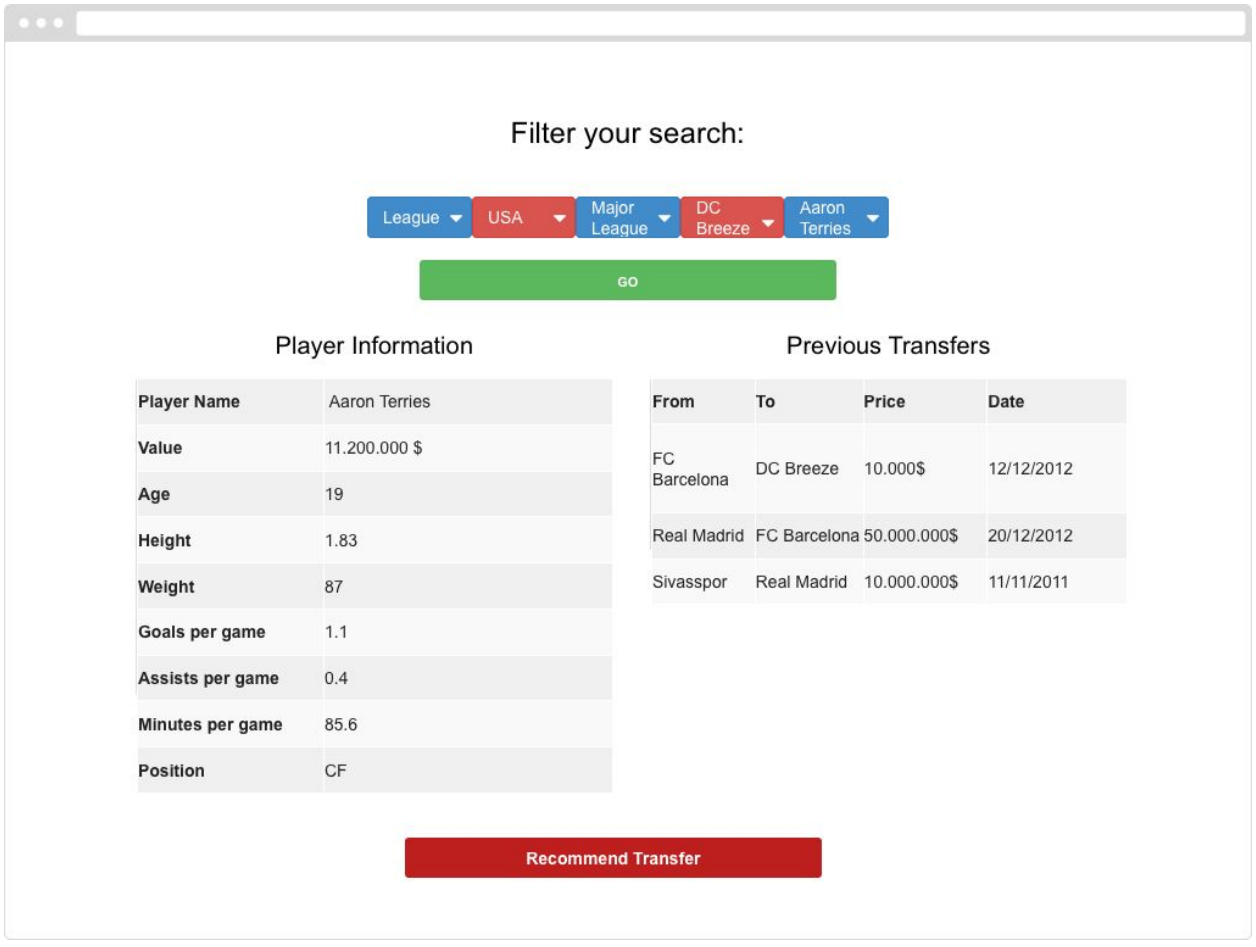


6.2-) Club Coach

Club Coach is also a regular user, and he has some extra features. If the user has a club coach account and he signs in, there will be differences in some pages.

6.2.1-) Player Details, Previous Transfers and Transfer Requests

In player information page, there is a button which is used for recommending transfer to the club manager. If a player plays at the same club with club coach, the club coach cannot recommend this player.

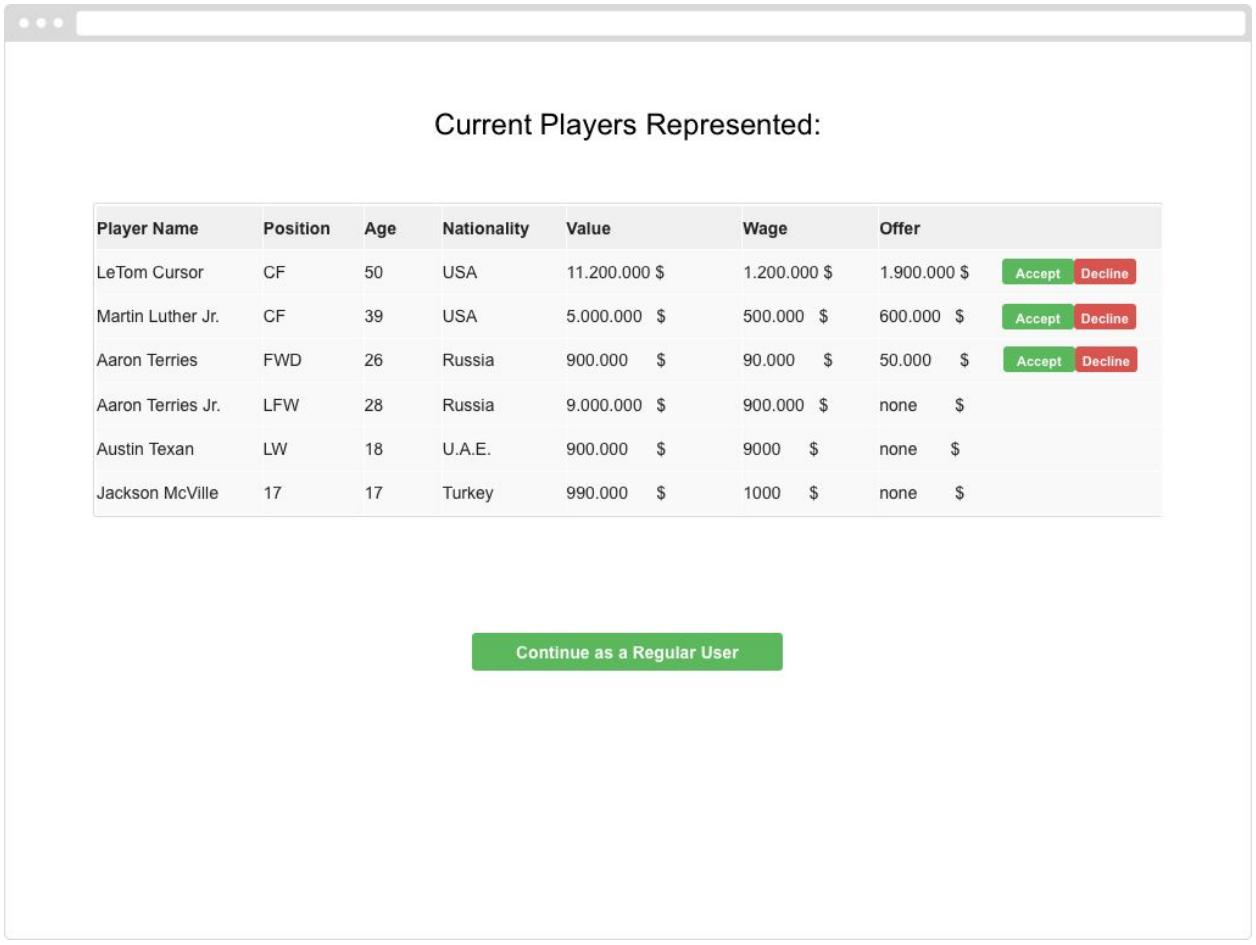


6.3-) Player Agent

Player agent is also a regular user, and he has some extra features. If the user has a player agent account and he signs in, there will be differences in some pages.

6.3.1-) Home page

The home page is different if the user is player agent. User can see all of the players he represents. Also, available contract offers ,which are made for these players by club managers of other clubs, are shown even if the club managers of these players accepts the transfer offer.



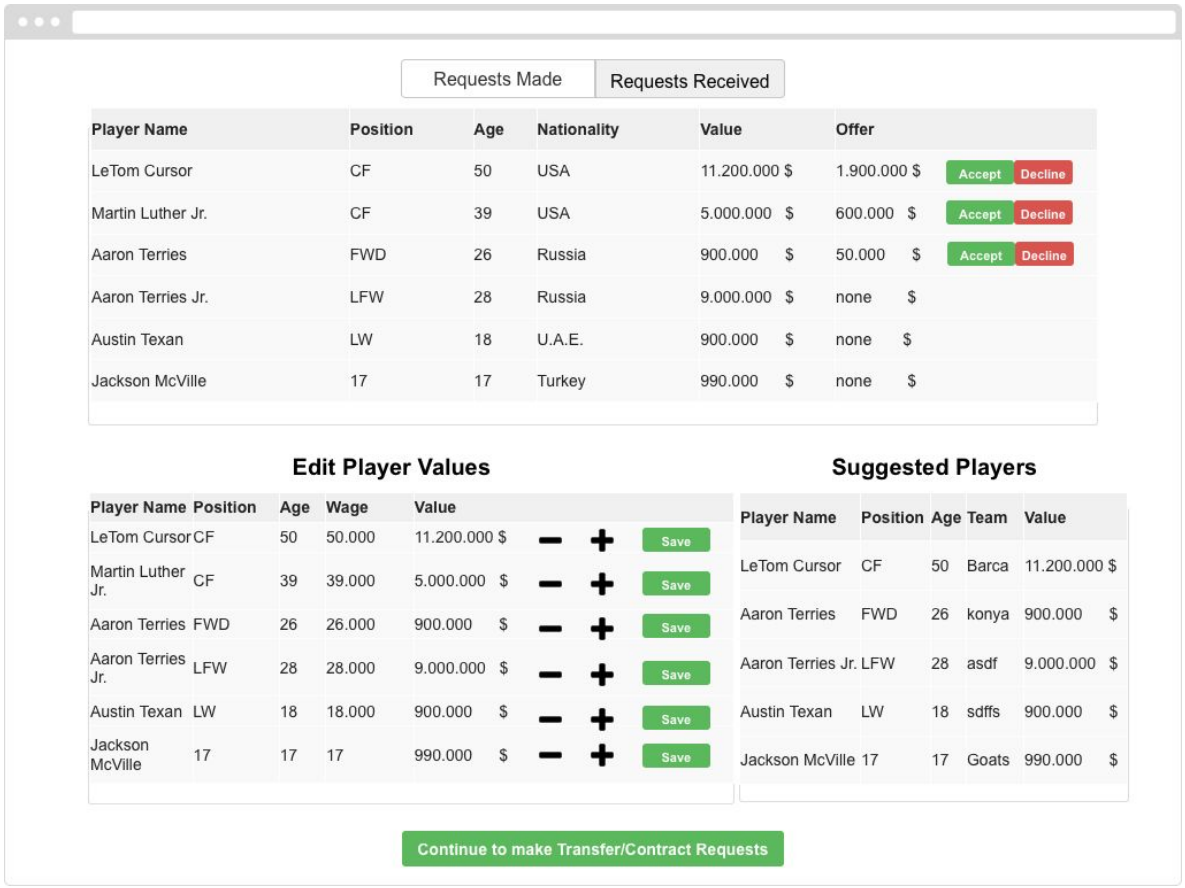
If the player agent clicks “Continue as a Regular User” button, he is redirected to home page of regular user.

6.4-) Club Manager

Club manager is also a regular user, and he has some extra features. If the user has a club manager account and he signs in, there will be differences in some pages.

6.4.1-) Home page

The home page is different if the user is club manager. User can see his entire team players and their information. Club manager can set a new value for the players in his team. Also, players suggested by club coach will be shown. The club manager can manage his transfer offers and the transfer offers that are come to his players.



If the club manager clicks “Continue as a Regular User” button, he is redirected to home page of regular user.

6.4.2-) Player Details, Previous Transfers and Sending Transfer Requests

In player information page, there is a button which is used for sending transfer request for the player to player's club manager. At the same time, he sends the contract offer for the player to player's player agent. Player agent of this player can only see this contract offer if the club manager of this player accepts the transfer offer. If a player plays at the club managed by the user, user cannot send request for this player.