Bilkent University

Department of Computer

Football Database System

CS-353 Group 10, Section 1

Project Design Report

Group Members:

- Solehjon Ruziboev
- Shamil Ibrahimov
- Mehmet Eren Turanboy
- Demir Topaktaş

1.Revised E/R Model

According to assistant's reviews, we revised our E/R model based on some of transfer points as follows:

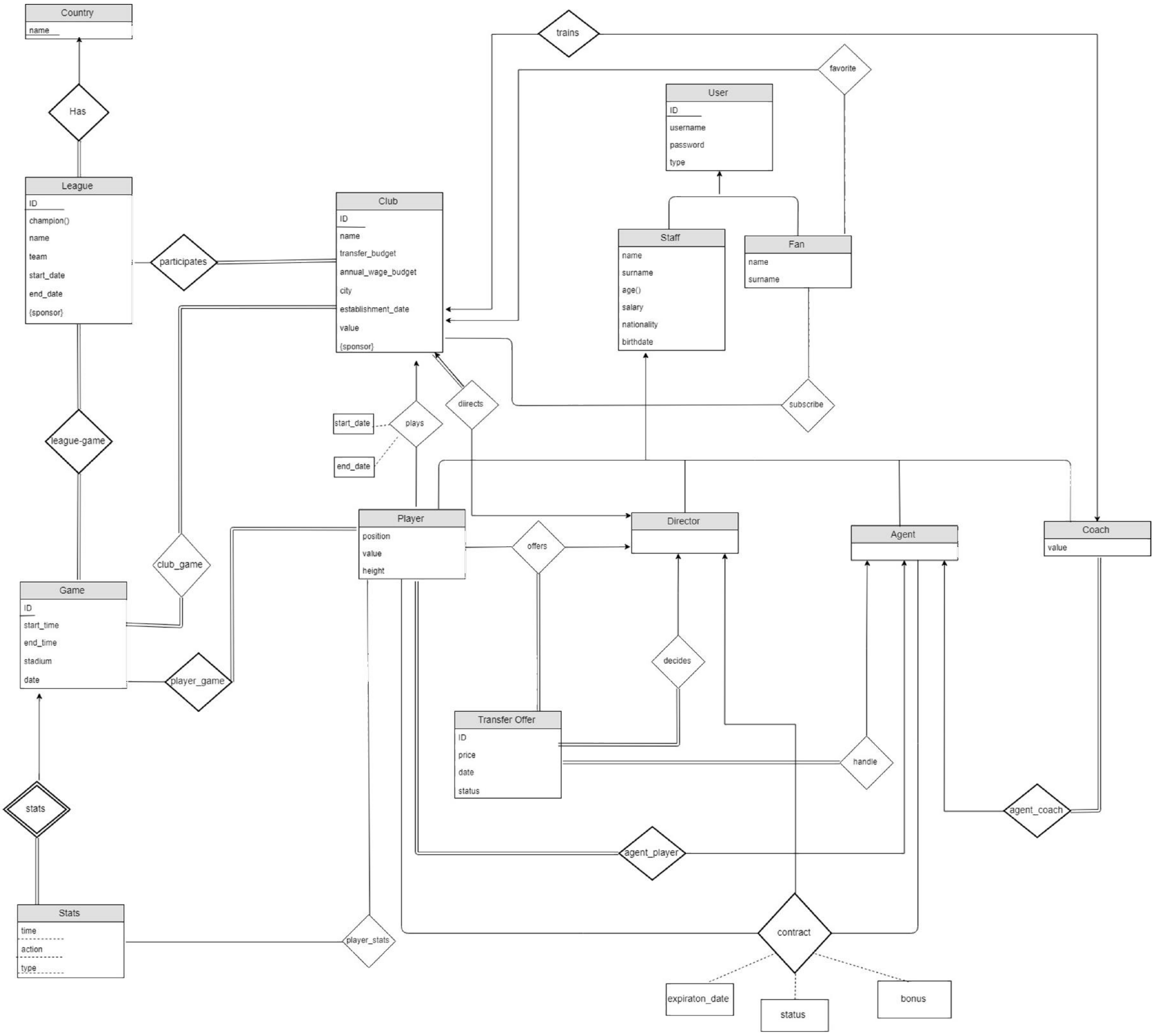
1. We changed offer relations to entity with attributes of ID, price, date, state. Then we made a ternary relations, which is offers, between Player, Agent and Director. Now, directors have a relation "decides" to decide the transfer offers and there is an agent

to handle this offers, as well. So directors and coaches are not the only only in the transfer offer procedure.

2. We merged Goal, Card, Assist week entities in to single Stats. Now it keeps the stats of Games and Players.
3. Coach cannot ask for a specific player’s transfer b deleting the wants relation.
4. Instead of writing IDs of entities near to relations we directly connect them to entities.
5. We added new additional features which are favorite team of the user and his/her subscribed teams.
6. Deleted past_clubs and current_club relations between Club and Player.
7. Deleted the attributes of club_league_stats since we can calculate them online.
8. We changed club_game relations to total participation.

We also made following changes in our E/R:

1. We made a User entity as a head of the ISA relation. Everyone inherits from User.
2. We added contract relation between Player and Agent entities.


Country
name
Has
League
ID
champion()
name
team
start_date
end_date
{sponsor}
participates
league-game
Game
ID
start_time
end_time
stadium
date
club_game
player_game
stats
Stats
time
action
type
player_stats
Club
ID
name
transfer_budget
annual_wage_budget
city
establishment_date
value
{sponsor}
trains
favorite
User
ID
username
password
type
Staff
name
surname
age()
salary
nationality
birthdate
Fan
name
surname
subscribe
diirects
plays
start_date
end_date
Player
position
value
height
offers
Director
decides
Transfer Offer
ID
price
date
status
handle
Agent
Coach
value
agent_coach
agent_player
contract
expiraton_date
status
bonus

2. Relation Schemas

2.1 Staff

Relational Model

Staff(ID, name, surname, age(), salary, nationality, birthdate)

Functional Dependencies

ID -> name, surname, age(), salary, nationality, birthdate

Candidate Keys

{(ID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Staff(
        ID              int NOT NULL PRIMARY KEY AUTO_INCREMENT,
        name            varchar(20) NOT NULL,
        surname         varchar(20) NOT NULL,
        age             int NOT NULL,
        salary          int NOT NULL,
        nationality     varchar(15) NOT NULL,
        birthdate       date NOT NULL,
        FOREIGN KEY (ID) REFERENCES User(ID)
);
```

2.2 Director

Relational Model

Director(ID, ClubID)

Functional Dependencies

ID -> ClubID

Candidate Keys

{(ID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Director(
        ID      int NOT NULL PRIMARY KEY AUTO_INCREMENT,
        ClubID int FOREIGN KEY REFERENCES Club(ID)
);
```

2.3 Agent

Relational Model

Agent(<u>ID</u>)

Functional Dependencies

No dependencies

Candidate Keys

{(ID))

Normal Form

BCNF

Table Definition

```
CREATE TABLE Agent(
        ID        int  NOT NULL PRIMARY KEY AUTO_INCREMENT
);
```

2.4 Player

Relational Model

Player(ID, position, value, height, AgentID)

Functional Dependencies

ID -> position, value, height, AgentID

Candidate Keys

{(ID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Player(
        ID               int NOT NULL PRIMARY KEY AUTO_INCREMENT,
        position        varchar(10) NOT NULL,
        value           int NOT NULL,
        height          int NOT NULL
        AgentID         int FOREIGN KEY REFERENCES Agent(ID)
);
```

2.5 Coach

Relational Model

Coach(ID, value, AgentID, ClubID)

Functional Dependencies

ID -> value, AgentID, ClubID

Candidate Keys

{(ID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Coach(
        ID              int NOT NULL PRIMARY KEY AUTO_INCREMENT,
        value           int NOT NULL,
        AgentID         int NOT NULL FOREIGN KEY REFERENCES Agent(ID),
        ClubID          int NOT NULL FOREIGN KEY REFERENCES Club(ID)
);
```

2.6 Country

Relational Model

Country(Name)

Functional Dependencies

No dependency

Candidate Keys

{(Name)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Country(
        Name  varchar(30) NOT NULL PRIMARY KEY,
);
```

2.7 League

Relational Model

League(ID, champion, name, start_date, end_date, CountryName)

Functional Dependencies

ID -> champion, name, start_date, end_date, CountryName

Candidate Keys

{(ID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE League(
        ID              int NOT NULL PRIMARY KEY AUTO_INCREMENT,
        champion        varchar(50) NOT NULL,
        name          varchar(20) NOT NULL,
        start_date    date NOT NULL,
        end_date      date NOT NULL,
        CountryName   varchar(30) FOREIGN KEY REFERENCES Country(name)
);
```

2.8 Game

Relational Model

Game(ID, start_time, end_time, stadium, date,home_teamID,away_teamID)

Functional Dependencies

ID -> stadium, start_time, end_time, date, home_teamID, away_teamID

Candidate Keys

{(ID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Game(
        ID            int NOT NULL PRIMARY KEY AUTO_INCREMENT,
        start_time    int NOT NULL,
        end_time      int NOT NULL,
        stadium       varchar(20) NOT NULL,
        gameDate        date NOT NULL
        home_teamID  int  NOT NULL FOREIGN KEY REFERENCES Club(ID)
        away_teamID  int  NOT NULL FOREIGN KEY REFERENCES Club(ID)
);
```

2.9 Stats

Relational Model

Stats(time, action, type, GameID, PlayerID)

Functional Dependencies

time, GameID -> action, type, PlayerID

Candidate Keys

{(time, GameID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Stats(
        time            int  NOT NULL,
        action          int NOT NULL,
        type            int NOT NULL,
        GameID          int FOREIGN KEY REFERENCES Game(ID),
        PRIMARY KEY (time, ID),
        PlayerID        int FOREIGN KEY REFERENCES Player(ID)
);
```

2.10 Fan

Relational Model

Fan(ID, name, surname, email, primaryPhone, favoriteTeamId)

Functional Dependencies

ID -> name, surname, email, primaryPhone, favoriteTeamId

Candidate Keys

{(ID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Fan(
        ID                  int NOT NULL PRIMARY KEY AUTO_INCREMENT,
        name                varchar(20) NOT NULL,
        surname             varchar(20) NOT NULL,
        email               varchar(30),
        primaryPhone        varchar(20),
        favoriteTeamId      int FOREIGN KEY REFERENCES Club(ID),
        FOREIGN KEY (ID) REFERENCES User(ID)
);
```

2.11 Club

Relational Model

Club(ID, name, transfer_budget, annual_wage_budget, city, establishment_date, value,stadium)

Functional Dependencies

ID -> name, transfer_budget, annual_wage_budget, city, establishment_date, value, stadium

Candidate Keys

{(ID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Club(
        ID   int NOT NULL PRIMARY KEY AUTO_INCREMENT,
        name varchar(20) NOT NULL,
        transfer_budget int NOT NULL,
        annual_wage_budget int NOT NULL,
        city varchar(20) NOT NULL,
        establishment_date date NOT NULL,
        value int NOT NULL,
        stadium varchar(20) NOT NULL
);
```

2.12 Transfer_Offer

Relational Model

Transfer_Offer(ID, price, transferDate, status, PlayerID, FromDirectorID, ToDirectorID)

Functional Dependencies

ID-> price, transferDate, status, PlayerID, FromDirectorID, ToDirectorID

Candidate Keys

{(ID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Transfer_Offer(
        ID                  int NOT NULL PRIMARY KEY AUTO_INCREMENT,
        price               int NOT NULL,
        transferDate        date NOT NULL,
        status              int NOT NULL,
        PlayerID            int NOT NULL FOREIGN KEY REFERENCES Player(ID),
        FromDirectorID      int NOT NULL FOREIGN KEY REFERENCES Director(ID),
        ToDirectorID        int NOT NULL FOREIGN KEY REFERENCES Director(ID)
);
```

2.13 League_Club

Relational Model

League_Club(LeagueID, ClubID)

Functional Dependencies

No Dependencies

Candidate Keys

{(LeagueID, ClubID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE League_Club(
        LeagueID   int NOT NULL,
        ClubID       int NOT NULL,
        PRIMARY KEY(LeagueID, ClubID),
        FOREIGN KEY(LeagueID) references League (ID),
        FOREIGN KEY(ClubID) references Club (ID)
);
```

2.14 League_Game

Relational Model

League_Game(LeagueID, GameID)

Functional Dependencies

No Dependencies

Candidate Keys

{(LeagueID, GameID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE League_Game(
        LeagueID   int NOT NULL,
        GameID     int NOT NULL,
        PRIMARY KEY(LeagueID, GameID),
        FOREIGN KEY(LeagueID) references League (ID),
        FOREIGN KEY(GameID) references Game (ID)
);
```

2.15 Player_Game

Relational Model

Player_Game(PlayerID, GameID)

Functional Dependencies

No Dependencies

Candidate Keys

{(PlayerID, GameID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Player_Game(

        PlayerID      int NOT NULL,

        GameID    int NOT NULL,

        PRIMARY KEY(PlayerID, GameID),

        FOREIGN KEY(PlayerID) references Player (ID),

        FOREIGN KEY(GameID) references Game (ID)

);
```

2.16 Plays

Relational Model

Plays(ClubID, PlayerID)

Functional Dependencies

No Dependencies

Candidate Keys

{(ClubID, PlayerID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Plays(

        ClubID      int NOT NULL,

        PlayerID    int NOT NULL,

        PRIMARY KEY(ClubID, PlayerID),

        FOREIGN KEY(PlayerID) references Player (ID),

        FOREIGN KEY(ClubID) references Club (ID)

);
```

2.17 Subscribe

Relational Model

Subscribe(<u>FanID, ClubID</u>)

Functional Dependencies

No Dependencies

Candidate Keys

{(FanID, ClubID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Subscribe(
        FanID       int NOT NULL,
        ClubID      int NOT NULL,
        PRIMARY KEY(FanID, ClubID),
        FOREIGN KEY(FanID) references Fan (ID),
        FOREIGN KEY(ClubID) references Club (ID)
);
```

2.18 Contract

Relational Model

Contract(PlayerID, DirectorID, AgentID, bonus, expiration_date, status)

Functional Dependencies

PlayerID, DirectorID, AgentID -> bonus, expiration_date, status

Candidate Keys

{(PlayerID, DirectorID, AgentID)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Contract(
        PlayerID                int NOT NULL,
        DirectorID              int NOT NULL,
        AgentID                 int NOT NULL,
        bonus                   int NOT NULL,
        expiration_date         date NOT NULL,
        status                  date NOT NULL,
        PRIMARY KEY(PlayerID, DirectorID, AgentID),
        FOREIGN KEY(PlayerID) references Player (ID),
        FOREIGN KEY(DirectorID) references Director (ID),
        FOREIGN KEY(AgentID) references Agent (ID)
);
```

2.19 League_Sponsor

Relational Model

League_Sponsor(LeagueID, SponsorName)

Functional Dependencies

No Dependencies

Candidate Keys

{(LeagueID, SponsorName)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE League_Sponsor(

        LeagueID        int NOT NULL,

        SponsorName  varchar(20) NOT NULL,

        PRIMARY KEY(LeagueID, SponsorName),

        FOREIGN KEY(LeagueID) references League (ID),

);
```

2.20 Club_Sponsor

Relational Model

Club_Sponsor(ClubID, SponsorName)

Functional Dependencies

No Dependencies

Candidate Keys

{(ClubID, SponsorName)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Club_Sponsor(
	ClubID		int NOT NULL,
	SponsorName	varchar(20) NOT NULL,
	PRIMARY KEY(ClubID, SponsorName),
	FOREIGN KEY(ClubID) references Club(ID)
);
```

2.21 User

Relational Model

User(ID, username, password, type)

Functional Dependencies

ID -> username, password, type

username -> ID, password, type

Candidate Keys

{(ID, username)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE User(
	ID		int NOT NULL PRIMARY KEY AUTO_INCREMENT,
        username        varchar(25) NOT NULL,

        password        varchar(25) NOT NULL,

        type            int NOT NULL,

        UNIQUE (username)

);
```

3. Functional Dependencies and Normalization Tables

Functional Dependencies are provided in section 2 of this report. All the relations are in Boyce Code Normal Form (BCNF). Thus no decomposition is required.

4. Functional Components

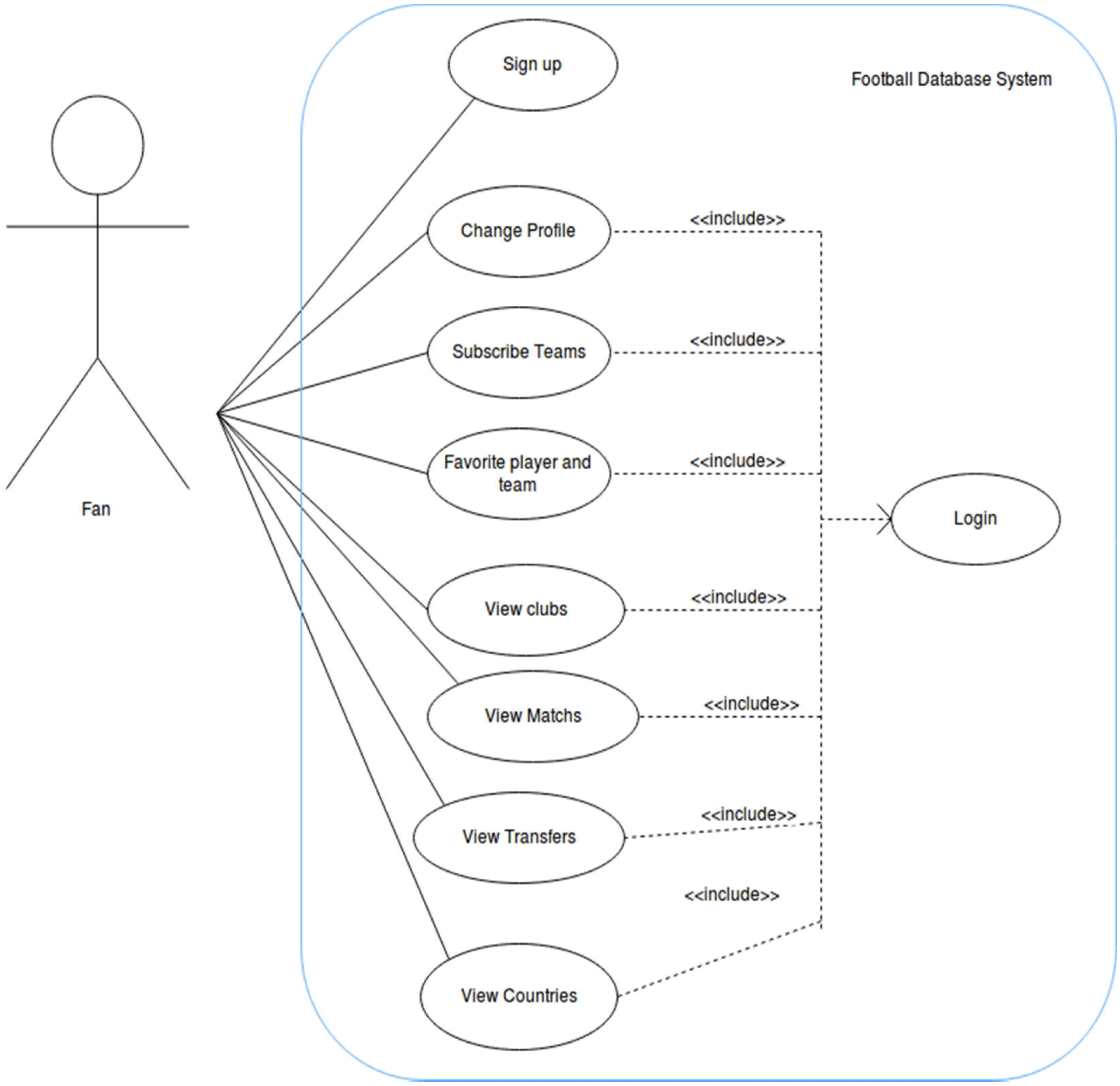
4.1 Use Cases

Football Database System provides functionalities to different user types. There are 7 users in the system which are Fan, Coach, Player, Director, Agent, Admin and Guest. Users share common functionalities and distinct functionalities.

4.1.1 Fan

- **Login:** Fan can login to the system by username and password. After login, fans can reach to their profile and have access to additional functionalities.
- **Sign up:** Fan can sign up to the system. Fan has to pick a unique username and unique password in order to create account.
- **Change profile:** Fans can change their profile. They can add additional informations such as address,age, nationality.
- **Subscribe teams:** Fans can subscribe to teams. When fan subscribed to a team, fan can be notified about matches of the team.

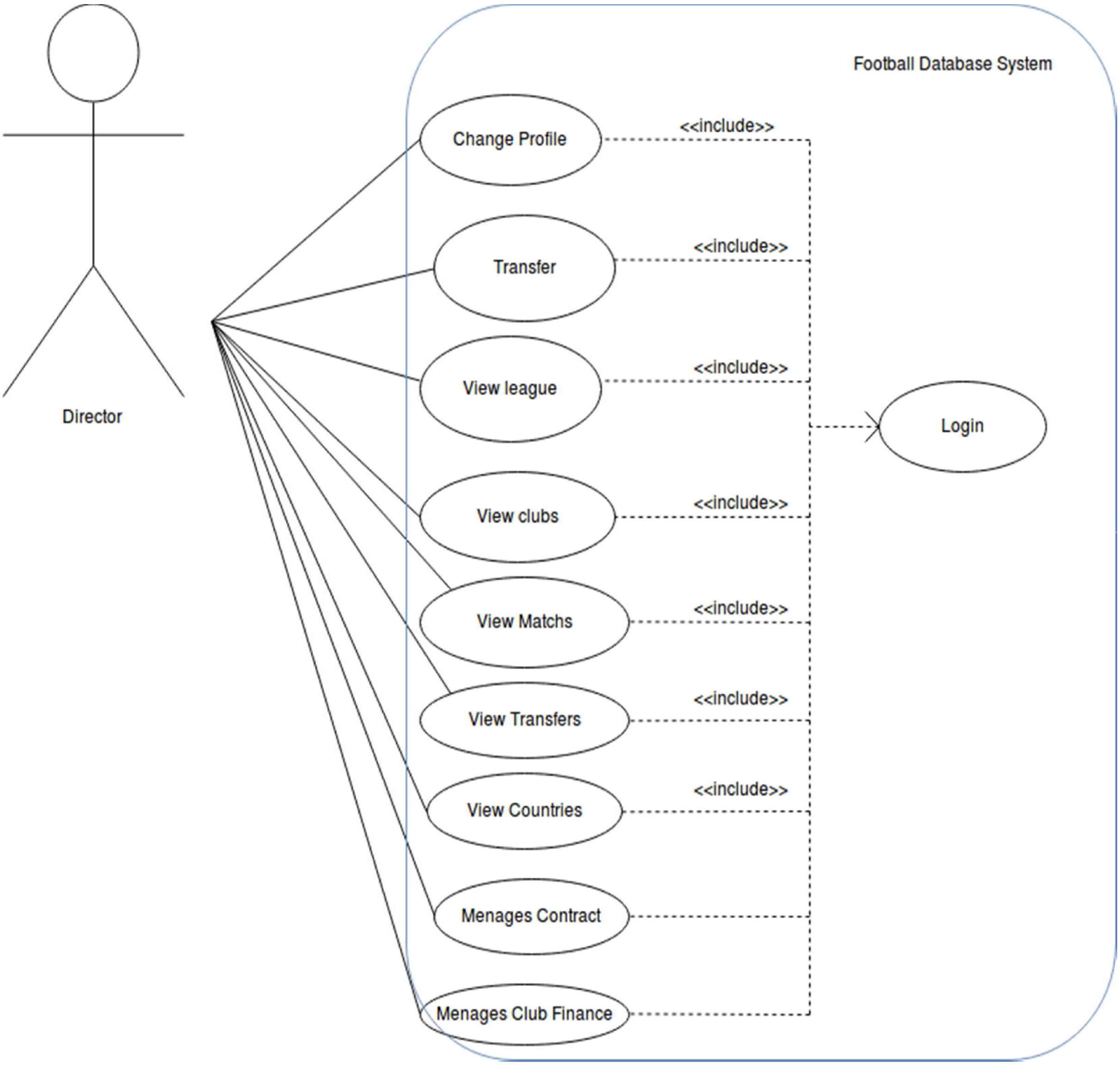
- **Favorite player and team:** Fans can choose the favorite team. When favorite player is chosen, fan’s profile will change and logo of the team will be displayed in fan’s profile.
- **View clubs:** Fans can search for teams and the system will provide information about team such as statistics ,team value ,standing in the league and information about club staff.
- **View leagues:** Fans can search for leagues and view information about leagues. Informations such as clubs that are participating in the league, league’s country, past winners of the league, previous standings of the clubs in the league will be displayed.
- **View matches:** Fans can search for past matches and view statistics about matches. Cards,goals and assists of the match can be viewed.
- **View transfers:** Fans can view transfers. Contract length, seller and buyer team, transfer fee, player that is transferred can be viewed by fan.
- **View countries:** Fans can view countries and countries leagues. Fan can see overall standing of the country, etc...



4.1.2 Director

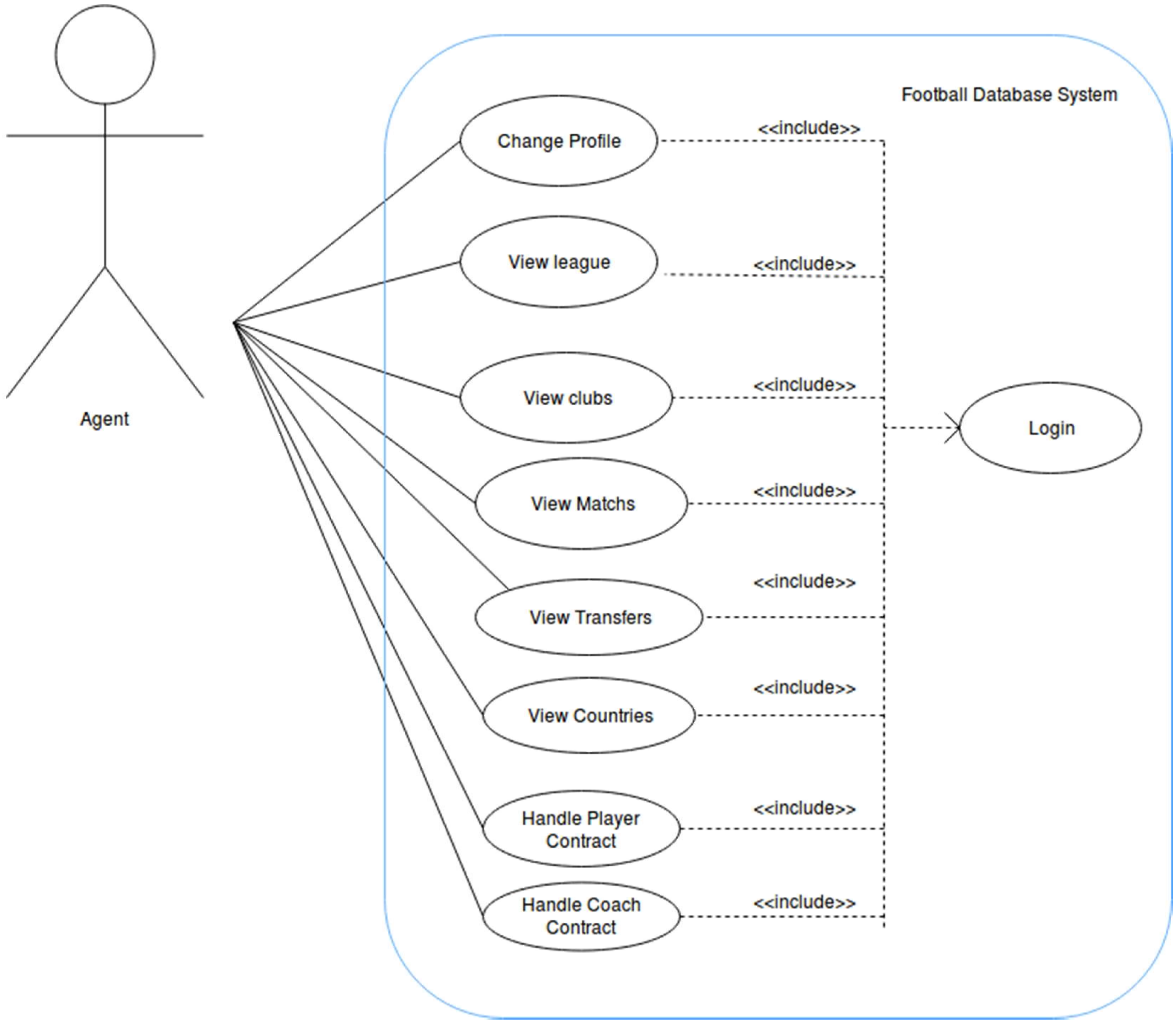
- **Login:** Director can login to the system by username and password. After login, directors can reach to their profile and have access to additional functionalities.
- **Change profile:** Director can change their profile. They can add additional informations such as address,age, nationality.

- **View clubs:** Directors can search for teams and the system will provide information about team such as statistics ,team value ,standing in the league and information about club staff.
- **View leagues:** Directors can search for leagues and view information about leagues. Informations such as clubs that are participating in the league, league’s country, past winners of the league, previous standings of the clubs in the league will be displayed.
- **View matches:** Directors can search for past matches and view statistics about matches. Cards,goals and assists of the match can be viewed.
- **View transfers:** Directors can view transfers. Contract length, seller and buyer team, transfer fee, player that is transferred can be viewed by fan.
- **View countries:** Directors can view countries and countries leagues. Director can see overall standing of the country, etc…
- **Transfer:** Directors can offer and evaluate the transfer offers from other directors.
- **Manages Contracts:** Directors can negotiate with agents to regulate contract.(bonuses,extending contract, etc..)
- **Manages club finances:** Directors can change expenses, incomes of the club.


Football Database System
Director
Change Profile
Transfer
View league
View clubs
View Matchs
View Transfers
View Countries
Menages Contract
Menages Club Finance
<<include>>
<<include>>
<<include>>
<<include>>
<<include>>
<<include>>
<<include>>
Login

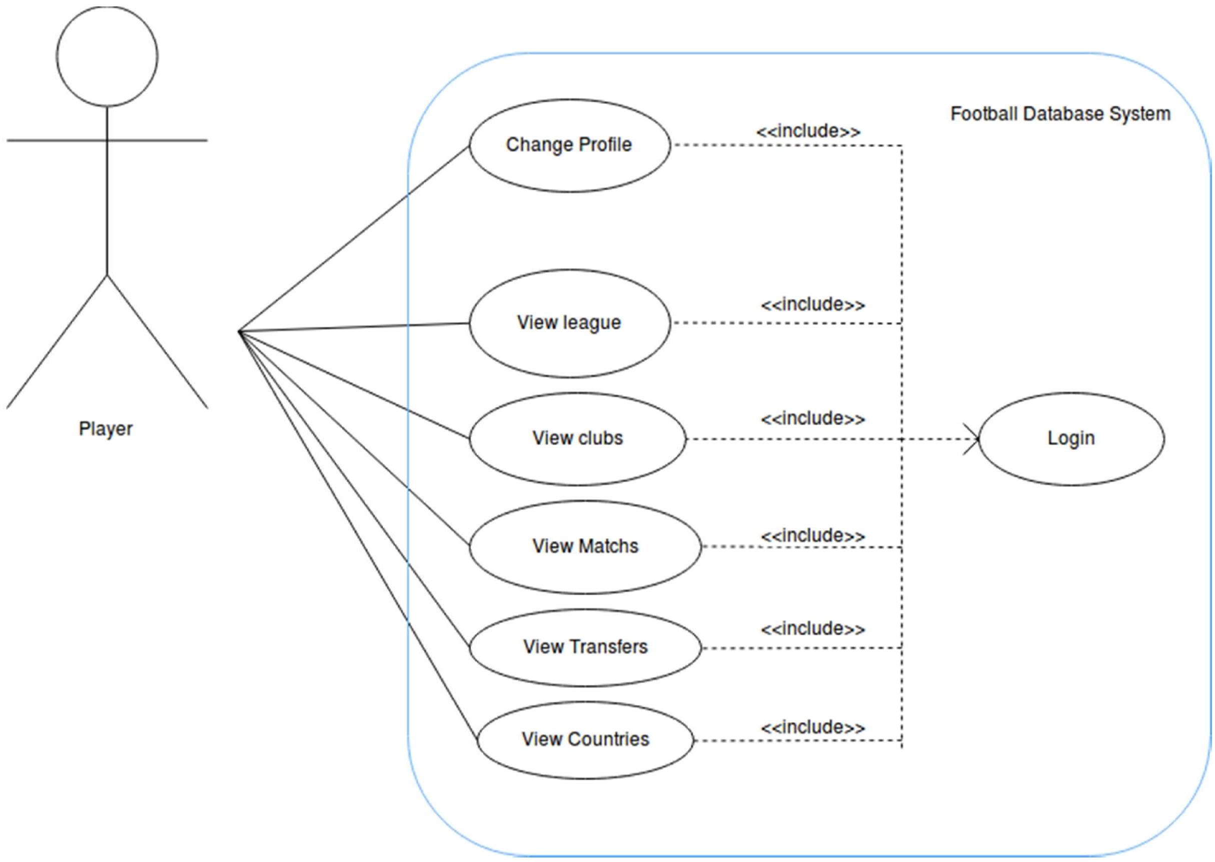
4.1.3 Agent

- **Login:** Agent can login to the system by username and password. After login, agents can reach to their profile and have access to additional functionalities.
- **Handle Coach Contract:** Agent can sign new contracts of the coaches whose contract is about the finish.
- **Handle Player Contract:** Agent can sign new contracts of the players whose contract is about the finish.
- **View clubs:** Agent can search for teams and the system will provide information about team such as statistics, team value, standing in the league and information about club staff.
- **View leagues:** Agent can search for leagues and view information about leagues. Informations such as clubs that are participating in the league, league's country, past winners of the league, previous standings of the clubs in the league will be displayed.
- **View matches:** Agent can search for past matches and view statistics about matches. Cards,goals and assists of the match can be viewed.
- **View transfers:** Agent can view transfers. Contract length, seller and buyer team, transfer fee, player that is transferred can be viewed by agent.
- **View countries:** Agents can view countries and countries leagues. Agents can see overall standing of the country, etc…


Agent
Football Database System
Change Profile
View league
View clubs
View Matchs
View Transfers
View Countries
Handle Player Contract
Handle Coach Contract
Login
<<include>>

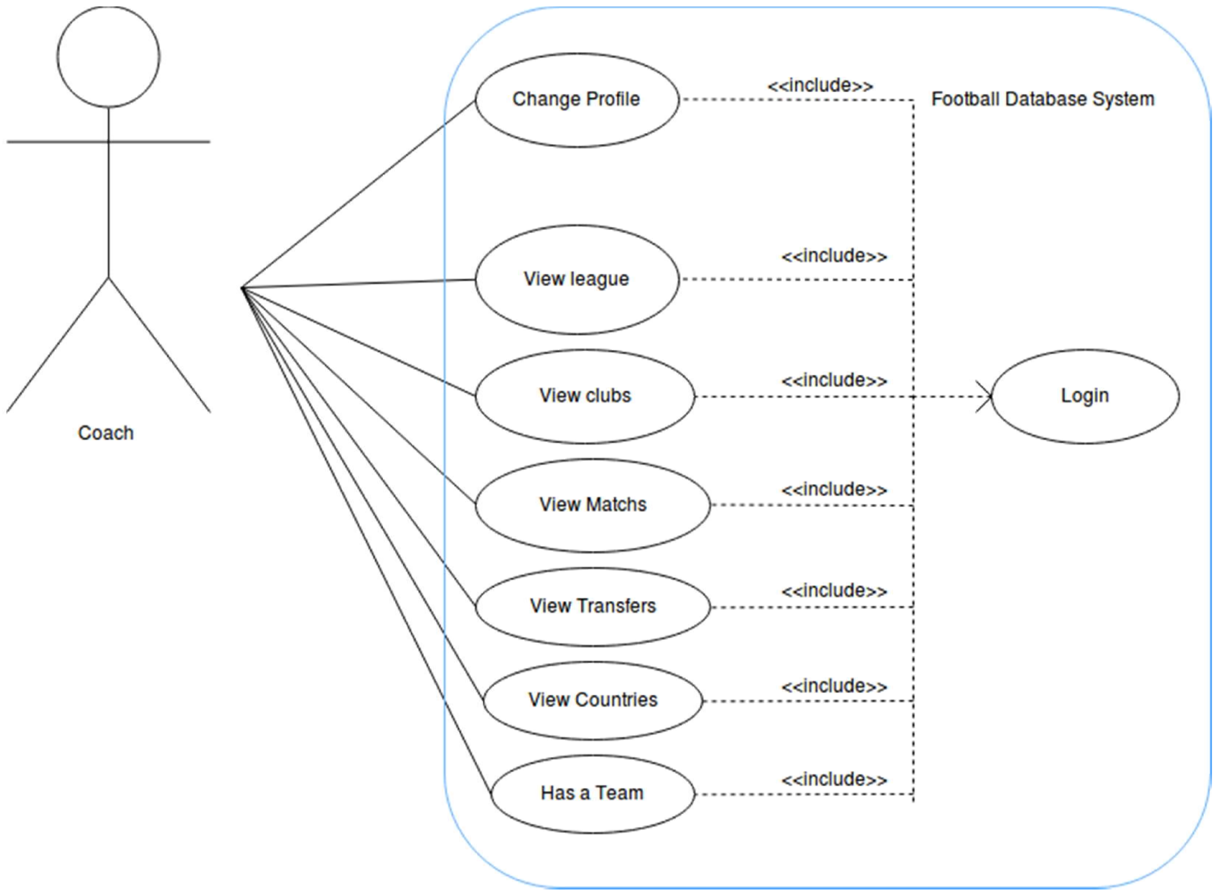
4.1.4 Player

- **Login:** Player can login to the system by username and password. After login, player can reach to their profile and have access to additional functionalities.
- **Change profile:** Player can change their profile. They can add additional informations such as address,age, nationality, club.
- **View clubs:** Player can search for teams and the system will provide information about team such as statistics ,team value ,standing in the league and information about club staff.
- **View leagues:** Player can search for leagues and view information about leagues. Informations such as clubs that are participating in the league, league's country, past winners of the league, previous standings of the clubs in the league will be displayed.
- **View matches:** Player can search for past matches and view statistics about matches. Cards,goals and assists of the match can be viewed.
- **View transfers:** Player can view transfers. Contract length, seller and buyer team, transfer fee, player that is transferred can be viewed by other players. Player can see only his transfer offers.
- **View countries:** Player can view countries and countries leagues. Player can see overall standing of the country, etc…


Player
Football Database System
Change Profile
View league
View clubs
View Matchs
View Transfers
View Countries
Login
<<include>>
<<include>>
<<include>>
<<include>>
<<include>>
<<include>>

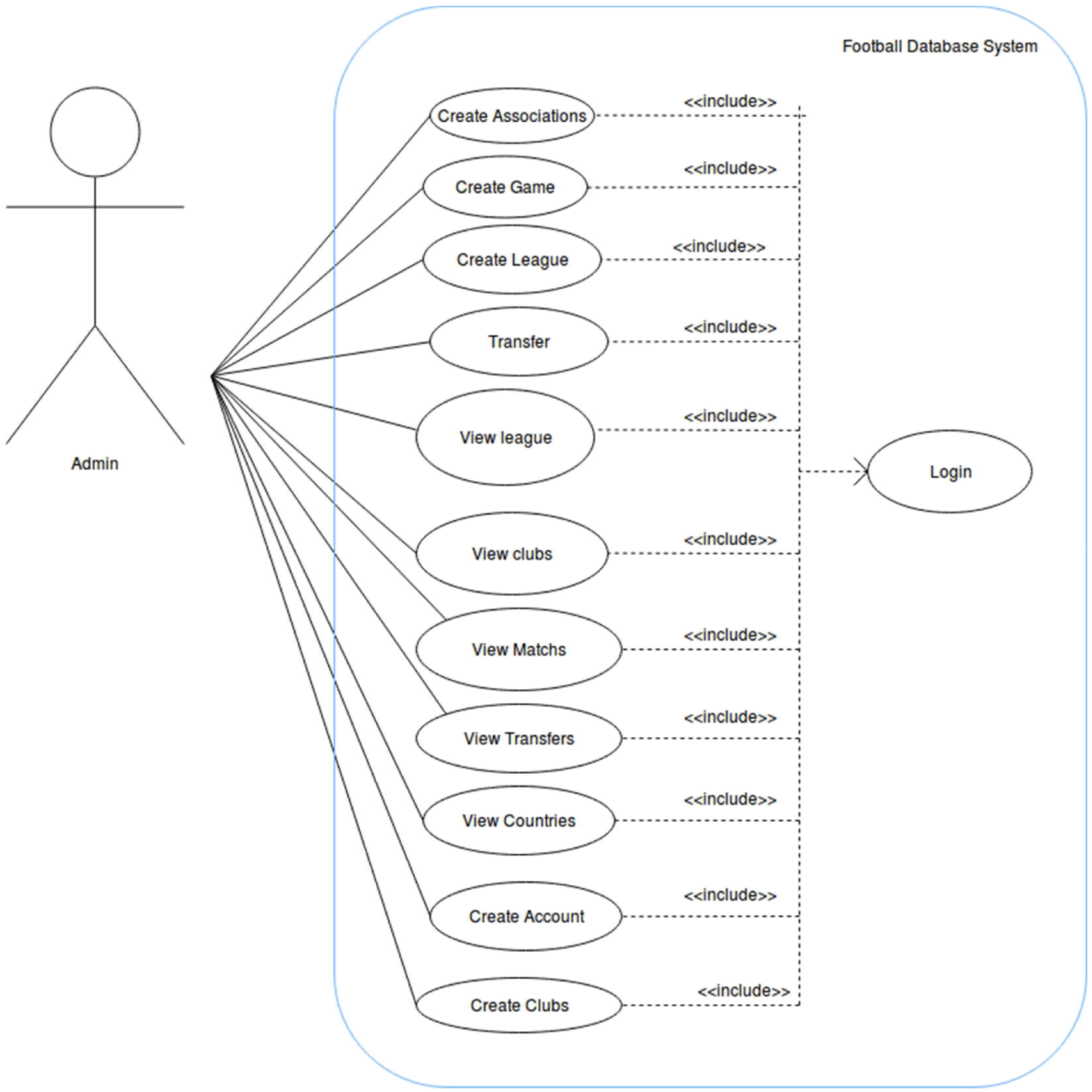
4.1.5 Coach

- **Login:** Coach can login to the system by username and password. After login, coaches can reach to their profile and have access to additional functionalities.
- **Change profile:** Coaches can change their profile. They can add additional informations such as address,age, nationality.
- **Has a team:** Coaches can train a team. In train team section, coach has access to club and player stats.
- **View clubs:** Coaches can search for teams and the system will provide information about team such as statistics ,team value ,standing in the league and information about club staff.
- **View leagues:** Coaches can search for leagues and view information about leagues. Informations such as clubs that are participating in the league, league’s country, past winners of the league, previous standings of the clubs in the league will be displayed.
- **View matches:** Coaches can search for past matches and view statistics about matches. Cards,goals and assists of the match can be viewed.
- **View transfers:** Coaches can view transfers. Contract length, seller and buyer team, transfer fee, player that is transferred can be viewed by fan.
- **View countries:** Coaches can view countries and countries leagues. Fan can see overall standing of the country, etc…


Coach
Change Profile
View league
View clubs
View Matchs
View Transfers
View Countries
Has a Team
<<include>>
Football Database System
Login

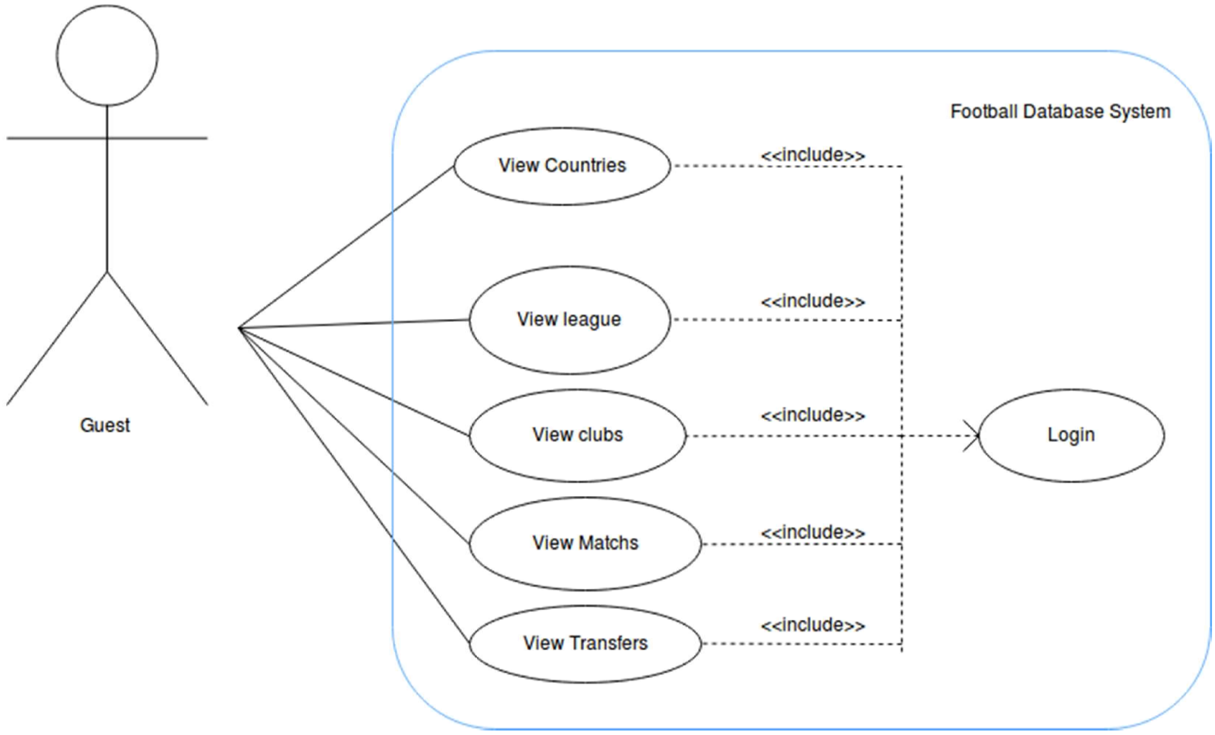
4.1.6 Admin

- **View clubs:** Admin can search for teams and the system will provide information about team such as statistics ,team value ,standing in the league and information about club staff.
- **View leagues:** Admin can search for leagues and view information about leagues. Informations such as clubs that are participating in the league, league’s country, past winners of the league, previous standings of the clubs in the league will be displayed.
- **View matches:** Admin can search for past matches and view statistics about matches. Cards,goals and assists of the match can be viewed.
- **View transfers:** Admin can view transfers. Contract length, seller and buyer team, transfer fee, player that is transferred can be viewed by fan.
- **View countries:** Admin can view countries and countries leagues. Fan can see overall standing of the country, etc…
- **Create Account:** Admin can create accounts for directors, coaches, players and agents.
- **Create Clubs:** Admin can create clubs.
- **Create Leagues:** Admin can create leagues.
- **Create Associations:** Admin can associate a player with club, player with agent, club with league, coach with club, league with game, club with game, league with country, player with game, league with sponsor, club with sponsor.
- **Create Games:** Admin can create games in a league.


Football Database System
Admin
Create Associations
Create Game
Create League
Transfer
View league
View clubs
View Matchs
View Transfers
View Countries
Create Account
Create Clubs
<<include>>
Login

4.1.7 Guest

- **View clubs:** Guest can search for teams and the system will provide information about team such as statistics ,team value ,standing in the league and information about club staff.
- **View leagues:** Guest can search for leagues and view information about leagues. Informations such as clubs that are participating in the league, league's country, past winners of the league, previous standings of the clubs in the league will be displayed.
- **View matches:** Guest can search for past matches and view statistics about matches. Cards,goals and assists of the match can be viewed.
- **View transfers:** Guest can view transfers. Contract length, seller and buyer team, transfer fee, player that is transferred can be viewed by fan.
- **View countries:** Guest can view countries and countries leagues. Fan can see overall standing of the country, etc…



4.2 Algorithms

Champion detection Algorithm

Each league has several teams competing with each other in order to win the league. As teams progress in the championship, they gather points, 3 for win, 1 for draw and 0 for lose. The league has fixed number of games, thus each club has an equal opportunity to gather points. In the end of the championship, when all the games have been played the total scores are summed up. The team with the most points is declared to be a champion. If the is more than one team with maximum points the number of goals will be a deciding factor.

In order to understand which team won in the game, the goals of both home and away teams are collected. In order to do so, each player's contribution from home team is added in the total goals of home team. The same is applied to the away team.

The total number of goals of a team is a sum of all goals of the club's players in a specific league.

Transfer Related Algorithms

In order to make a transfer of a player from one team to another several steps are needed to execute. The first thing, director needs to check his club's transfer budget and insure that it is enough to buy a player. The second step is for director to request a certain player from another director. The directors can access their offers and requests of players and their statuses online. In order to accept the offer they have to click the accept option of the certains request.

Now the agent has to handle the transfer. He can see his players and coaches transfer offers and can decide accordingly. If the agent accepts the offer the transfer will be successful otherwise the transfer won't happen. The player has a right to play in the team until the end of expiration date in the contract.

Contract Extension Algorithm

Players' and coaches' agent can request from the club in which the player/coach is playing/training. He sends a request to a director with the new conditions. The conditions might differ in bonus or more commonly in expiration date. All the contracts have a status attribute which shows the current stage of a contract, it can be accepted, rejected or pending.

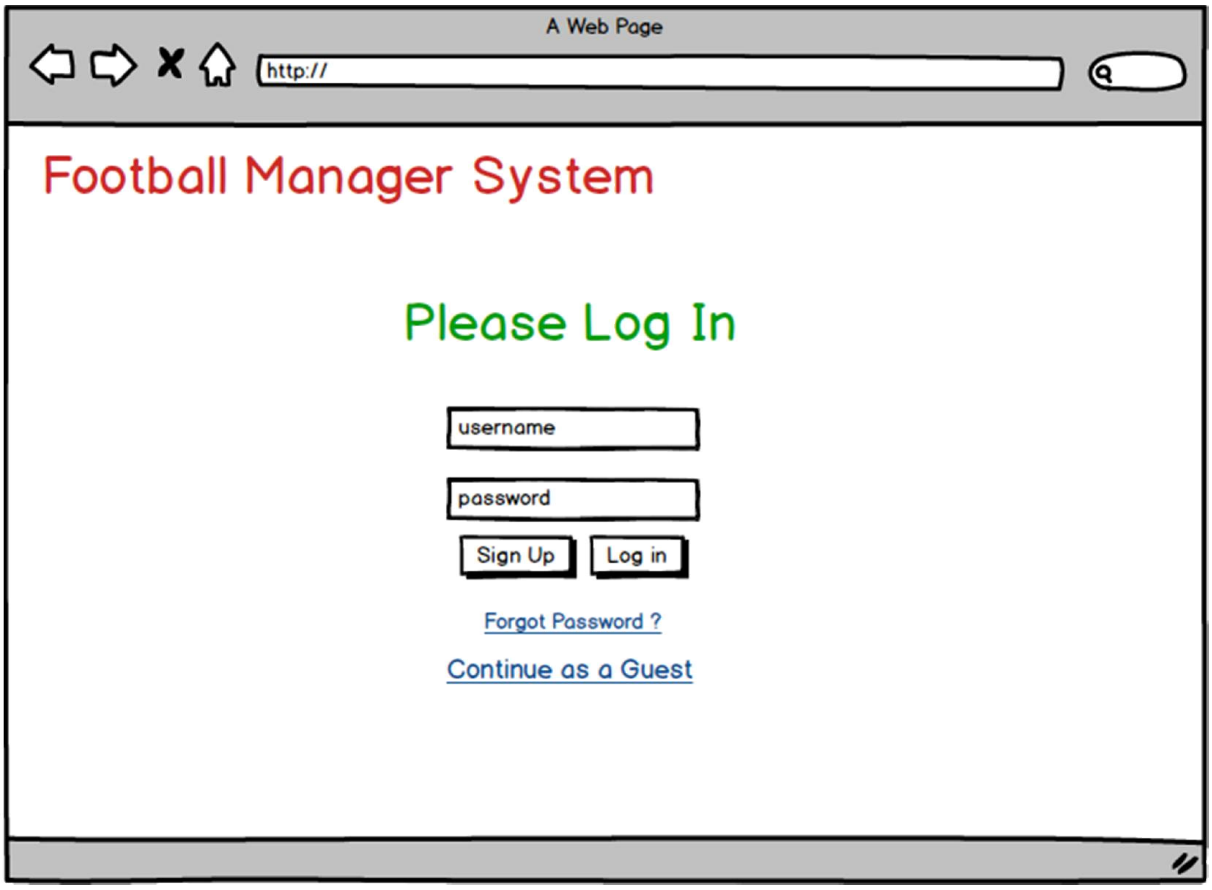
The director plays a main role in the contract extension. He can see all the extension requests in a separate view with the player, the agent and the new contract details. In the end he should decide whether to accept or decline the request.

4.3 Data Structures

The attributes of all the entities and relations of the current system are MySQL integers (int), dates (date), time (time) and strings (varchar).

5. User Interface and Corresponding SQL Statements

5.1 Login Page



Inputs: @username, @password

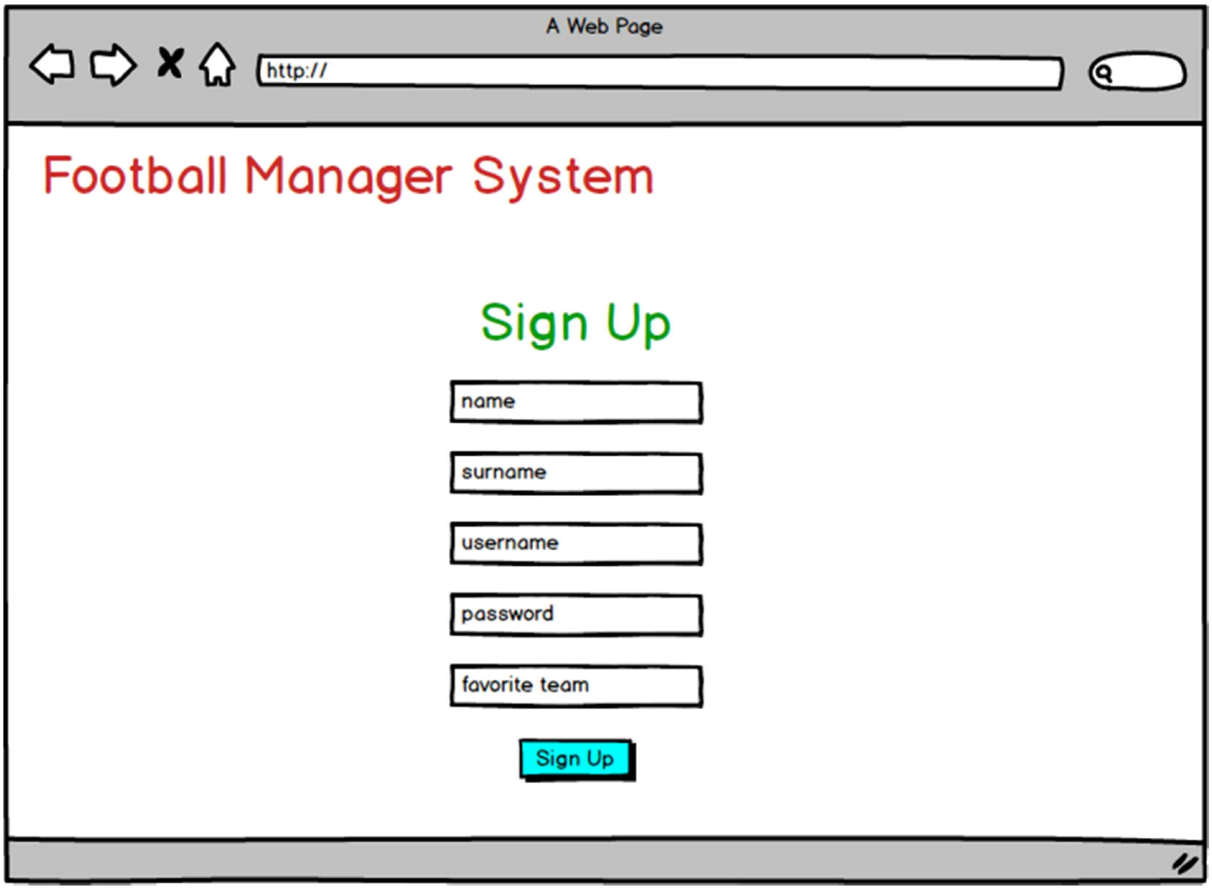
Process: This page allows fans to login after entering his /her username then clicking login button and password or if the fan forget he/she can click to link forgot password button to get help. User without an account can sign in by clicklicking sign up button after entering username and password or he/she can choose to enter the system by clicking continue as a guest link.

SQL Statements:

Retrieving a user

```
SELECT *
FROM user
WHERE user.username = @username AND user.password = @password
```

5.2 Sign-Up Page



Input: @name, @surname, @username, @password, @favoriteTeamID, @newFanID

Process: In this page user can sign up. This page is for creating new fan. In order to sign up user should enter his/her name, surname, username password and favorite team's name. This page is accessible for every user.

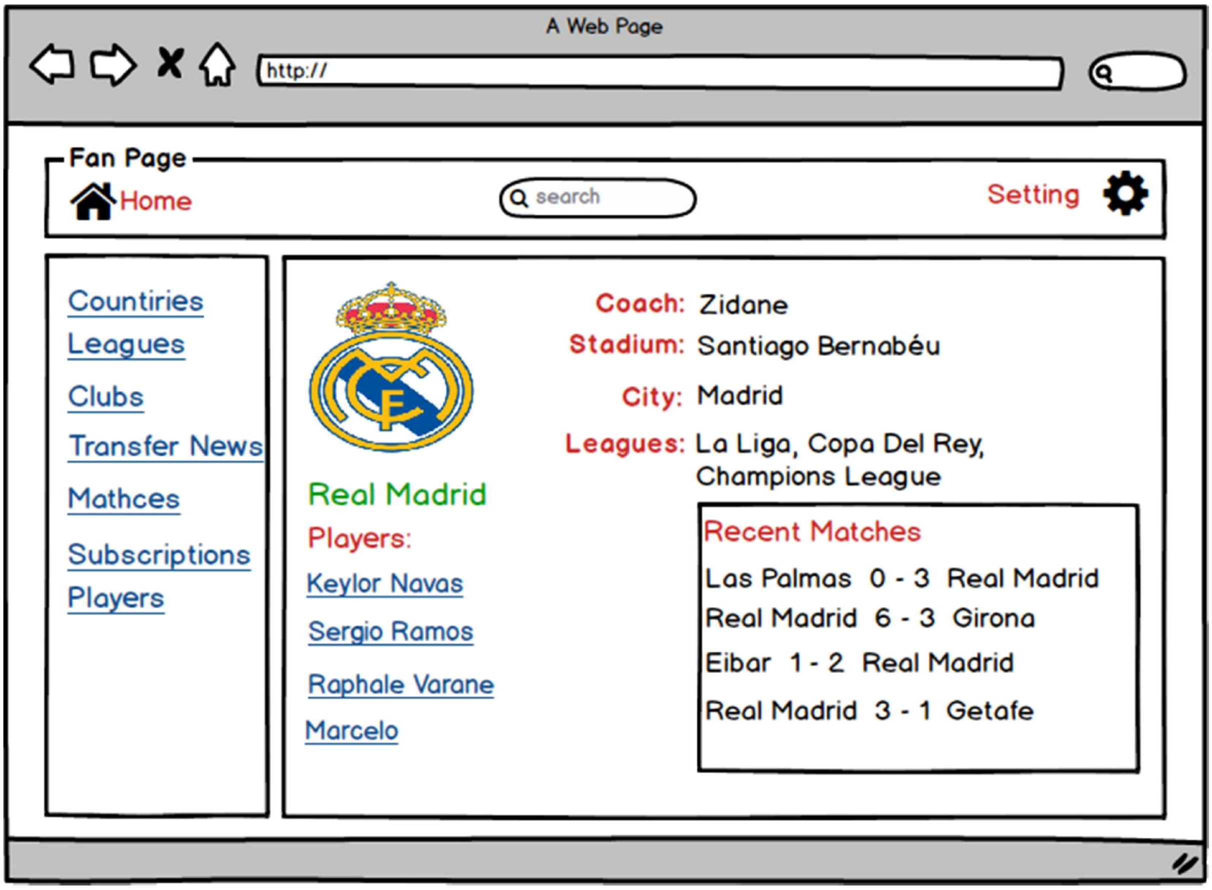
SQL Statements:

Adding New Fan:

```
INSERT INTO Fan
values(@newFanID, @name, @surname, @favoriteTeamID);


INSERT INTO User
values(@newFanID, @username, @password, ‘fan’);
```

5.3 Fan Page



Real Madrid

Players:

Keylor Navas

Sergio Ramos

Raphale Varane

Marcelo

Coach: Zidane

Stadium: Santiago Bernabéu

City: Madrid

Leagues: La Liga, Copa Del Rey, Champions League

Recent Matches

Las Palmas 0 - 3 Real Madrid

Real Madrid 6 - 3 Girona

Eibar 1 - 2 Real Madrid

Real Madrid 3 - 1 Getafe

Input: @searchText, @curUserID

Process: In fan page there is a home button at the top of the page and it send fan to homepage which is actually current page in which includes favorite team's information such as club amblem, coach, stadium, city, currently playing leagues, recent matches of the favorite club and players of the favorite team. Moreover at the top of the page there is a search box which is to entering keyword to search. Settings button sends user to setting page. In the left side of the page there are links that sends user to pages that shows all countries, leagues, clubs, transfer news, matches, subscriptions and players. This page is only accessible for fans.

SQL Statements:

Retrieving User's Favorite Team:

```
SELECT Fan.favoriteTeamID

FROM Fan

WHERE Fan.ID = @curUserID
```

Retrieving Favorite Team's Info:

```
with favTeam(ID) as (
        SELECT Fan.favoriteTeamID
        FROM Fan
        WHERE Fan.ID = @curUserID
        )
        SELECT Club.name, Club.stadium, Club.city
        FROM Club, favTeam
        WHERE Club.ID = favTeam.ID
```

Retrieving Favorite Team's Coach:

```
with favTeam(ID) as (
        SELECT Fan.favoriteTeamID
        FROM Fan
        WHERE Fan.ID = @curUserID
        )
        SELECT Coach.name
        FROM Coach, favTeam
        WHERE Coach.clubID = favTeam.ID
```

Retrieving Favorite Team's Leagues:

```
with favTeam(ID) as (
        SELECT Fan.favoriteTeamID
        FROM Fan
        WHERE Fan.ID = @curUserID
        )
        SELECT League.name
        FROM Leage_Club, League, favTeam
        WHERE League_Club.LeagueID = League.ID AND Leage_Club.ClubID = favTeam.ID
```

Retrieving Favorite Team’s Players:

```
with favTeam(ID) as (
        SELECT Fan.favoriteTeamID
        FROM Fan
        WHERE Fan.ID = @curUserID
        )
        SELECT Player.name, Player.surname
        FROM plays, Player, favTeam
        WHERE plays.PlayerID = Players.ID AND favTeam.ID = plays.ClubID
```

Retrieving Favorite Team’s Recent Matches:

```
with favTeam(ID) as (
        SELECT Fan.favoriteTeamID
        FROM Fan
        WHERE Fan.ID = @curUserID
        )
with games(ID, homeID, awayID, homeTeamName, awayTeamName) as(
        SELECT Game.ID, Game.home_teamID, Game.away_teamID, Club1.name,
Club2.name
        FROM (SELECT * FROM Game ORDER BY Game.date DESC) G, Club Club1, Club
Club2
        WHERE Game.home_teamID = Club1.ID AND Game.away_teamID = Club2.ID AND
(G.home_teamID = favTeamID OR G.away_teamID = favTeamID)
)
with homeGoals(ID, no) as(
        SELECT games.homeID, count(*)
        FROM games, stats, plays
        WHERE games.ID = stats.GameID AND plays.PlayerID = stats.PlayerID AND
        plays.ClubID = games.homeID AND stats.action = ‘goal’
        )
with awayGoals(ID, no) as(
        SELECT games.awayID, count(*)
        FROM games, stats, plays
        WHERE games.ID = stats.GameID AND plays.PlayerID = stats.PlayerID AND
        plays.ClubID = games.awayID AND stats.action = ‘goal’
        )
SELECT games.homeTeamName as homeTeamName, games.awayTeamName as
awayTeamName,  homeGoals.no as homeGoals,  awayGoals.no as awayGoals
  FROM games, homeGoals, awayGoals
  WHERE games.homeID = homeGoals.ID AND games.awayID = awayGoals.ID
```

General Search SQL:

```
   SELECT Club.name, Club.city, Club.value
   FROM Club
   WHERE  Club.name = @user_input
```

```
  SELECT Player.name, Player.value, Player.salary
  FROM Player
  WHERE  Player.name = @user_input
```

```
  SELECT  League.Name
  FROM    League
  WHERE  League.countryName = @user_input
```

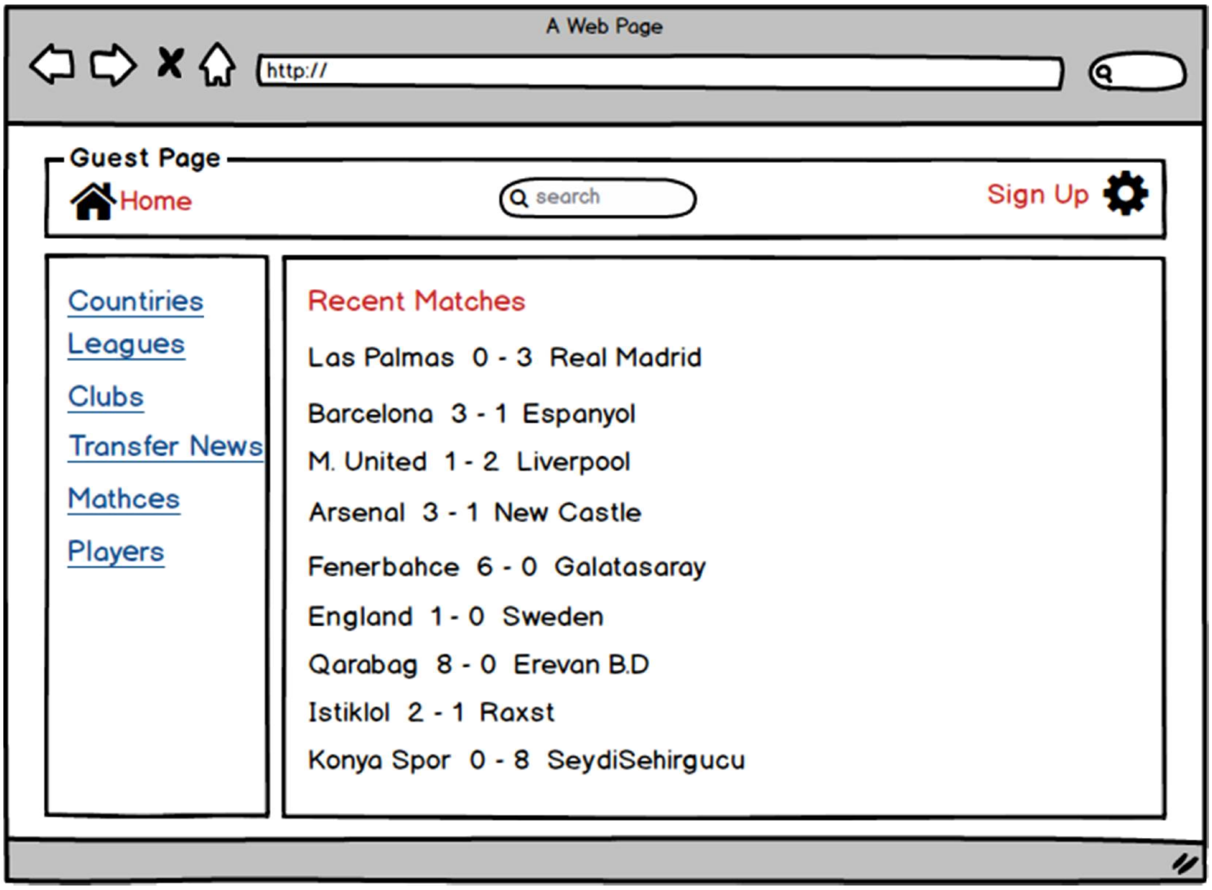
```
SELECT  League.Name
FROM     League
WHERE  League.Name = @user_input
```

```
SELECT club1.name,  matches.homeGoals, club2.name, matches.awayGoals
FROM     matches, club club1 , club club2
WHERE   matches.homeID = club1.ID AND matches.awayID = club2.ID
           AND ( club1.name = @user_input OR club2.name = @user_input)
```

```
  CREATE TABLE  matches AS
  with games(ID, homeID, awayID, homeTeamName, awayTeamName) as(
       SELECT Game.ID, Game.home_teamID, Game.away_teamID, Club1.name,
Club2.name
       FROM Game, Club Club1, Club Club2
       ORDER BY Game.date DESC
       WHERE Game.home_teamID = Club1.ID AND Game.away_teamID = Club2.ID
)
 with homeGoals(ID,no) as(
       SELECT games.homeID, count(*)
       FROM games, stats, plays
       WHERE games.ID = stats.GameID AND plays.PlayerID = stats.PlayerID AND
       plays.ClubID = games.homeID AND stats.action = ‘goal’
       )
 with awayGoals(ID,no) as(
       SELECT games.awayID,count(*)
       FROM games, stats, plays
```

```
        WHERE games.ID = stats.GameID AND plays.PlayerID = stats.PlayerID AND
        plays.ClubID = games.awayID AND stats.action = ‘goal’
        )
SELECT games.homeTeamName as homeTeamName, games.awayTeamName as
awayTeamName,  homeGoals.no as homeGoals,  awayGoals.no as awayGoals
    FROM games, homeGoals, awayGoals
    WHERE games.homeID = homeGoals.ID AND games.awayID = awayGoals.ID
```

5.4 Guest Page



Recent Matches

Las Palmas 0 - 3 Real Madrid

Barcelona 3 - 1 Espanyol

M. United 1 - 2 Liverpool

Arsenal 3 - 1 New Castle

Fenerbahce 6 - 0 Galatasaray

England 1 - 0 Sweden

Qarabag 8 - 0 Erevan B.D

Istiklol 2 - 1 Raxst

Konya Spor 0 - 8 SeydiSehirgucu

Input: @searchText

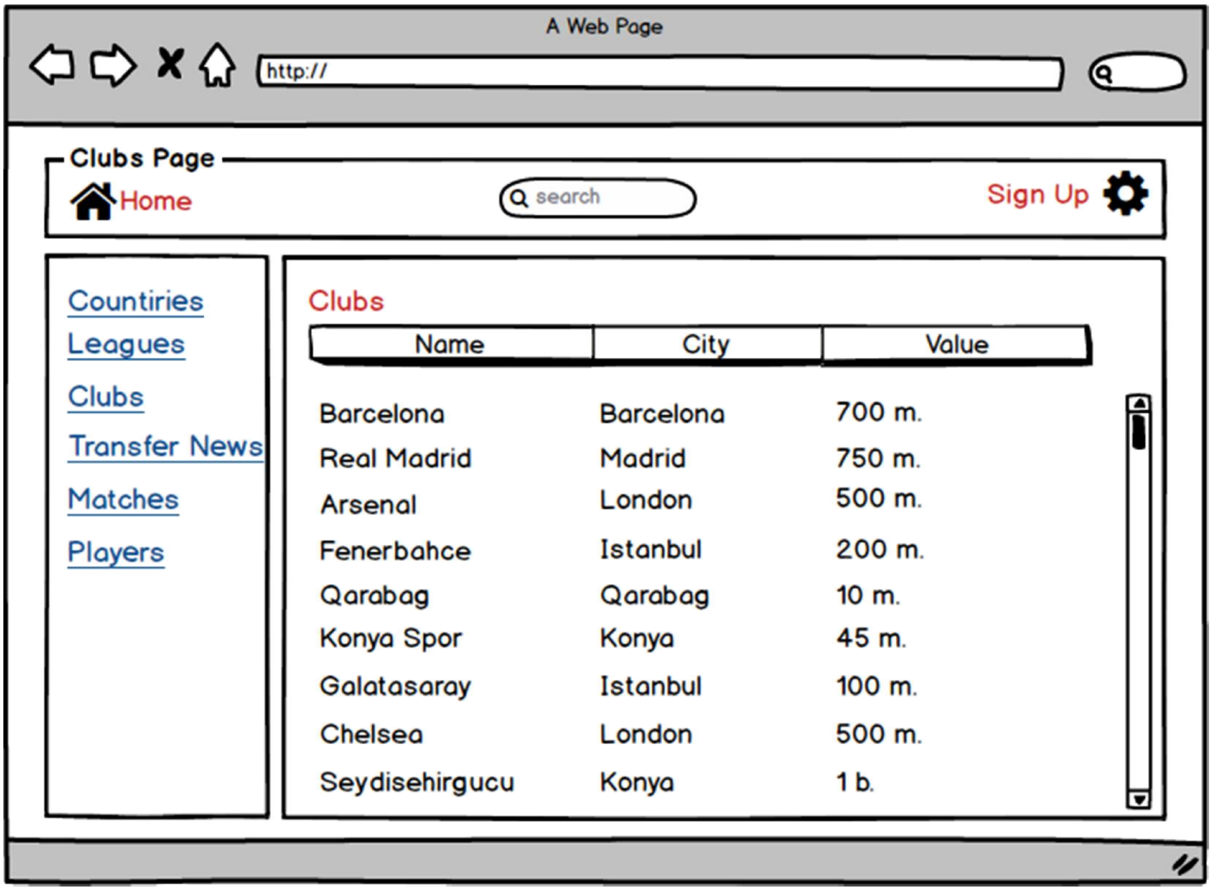
Process: In guest page there is a list of the recent matches that has been entered in football database system. The home button at the top of the page refreshes the current page. In addition, at the top of the page there is a search box which is to entering keyword to search. sign up button sends guest to login page. In the left side of the page there are links that sends guest to pages that shows all countries, leagues, clubs, transfer news, matches and players. The guests will be able to access this page.

SQL Statements:

```
with games(ID, homeID, awayID, homeTeamName, awayTeamName) as(
        SELECT Game.ID, Game.home_teamID, Game.away_teamID, Club1.name, 
Club2.name
        FROM Game, Club Club1, Club Club2
        ORDER BY Game.date DESC
        WHERE Game.home_teamID = Club1.ID AND Game.away_teamID = Club2.ID
)
  with homeGoals(ID,no) as(
        SELECT games.homeID, count(*)
        FROM games, stats, plays
        WHERE games.ID = stats.GameID AND plays.PlayerID = stats.PlayerID AND
        plays.ClubID = games.homeID AND stats.action = ‘goal’
        )
  with awayGoals(ID,no) as(
        SELECT games.awayID,count(*)
        FROM games, stats, plays
        WHERE games.ID = stats.GameID AND plays.PlayerID = stats.PlayerID AND
        plays.ClubID = games.awayID AND stats.action = ‘goal’
        )
SELECT games.homeTeamName as homeTeamName, games.awayTeamName as
awayTeamName,  homeGoals.no as homeGoals,  awayGoals.no as awayGoals
   FROM games, homeGoals, awayGoals
   WHERE games.homeID = homeGoals.ID AND games.awayID = awayGoals.ID
```

The Search in this page will lead to general Search written in Section 5.3

5.5 Clubs Page



Name	City	Value
Barcelona	Barcelona	700 m.
Real Madrid	Madrid	750 m.
Arsenal	London	500 m.
Fenerbahce	Istanbul	200 m.
Qarabag	Qarabag	10 m.
Konya Spor	Konya	45 m.
Galatasaray	Istanbul	100 m.
Chelsea	London	500 m.
Seydisehirgucu	Konya	1 b.

Input: @searchText

Process: In clubs page all the clubs that are entered in database system are listed. In the list clubs name their cities and the clubs value is visualized. Moreover it has all the features of the guest page. This page is accessible for every user.

SQL Statements:

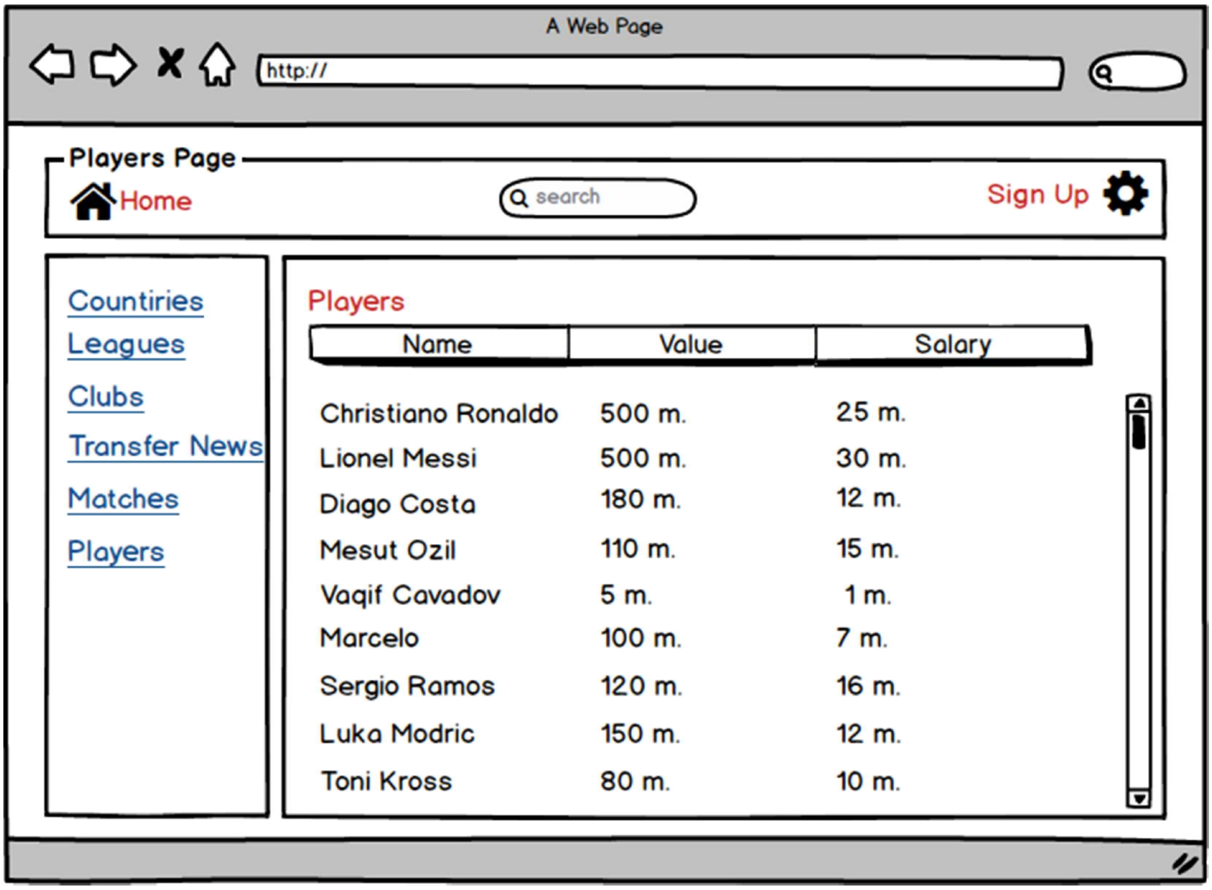
Retrieving Clubs Info:

```
SELECT Club.name, Club.city, Club.value
FROM Club
```

Searching in Clubs:

```
SELECT Club.name, Club.city, Club.value
FROM Club
WHERE Club.name = @searchText
```

5.6 Players Page



Players

Name	Value	Salary
Christiano Ronaldo	500 m.	25 m.
Lionel Messi	500 m.	30 m.
Diago Costa	180 m.	12 m.
Mesut Ozil	110 m.	15 m.
Vaqif Cavadov	5 m.	1 m.
Marcelo	100 m.	7 m.
Sergio Ramos	120 m.	16 m.
Luka Modric	150 m.	12 m.
Toni Kross	80 m.	10 m.

Input: @searchText

Process: In players page all the players that are entered in database system are listed. In the list players' name and surname their values and the players' salaries are visualized. Moreover it has all the features of the guest page. This page is accessible for every user.

SQL Statements:

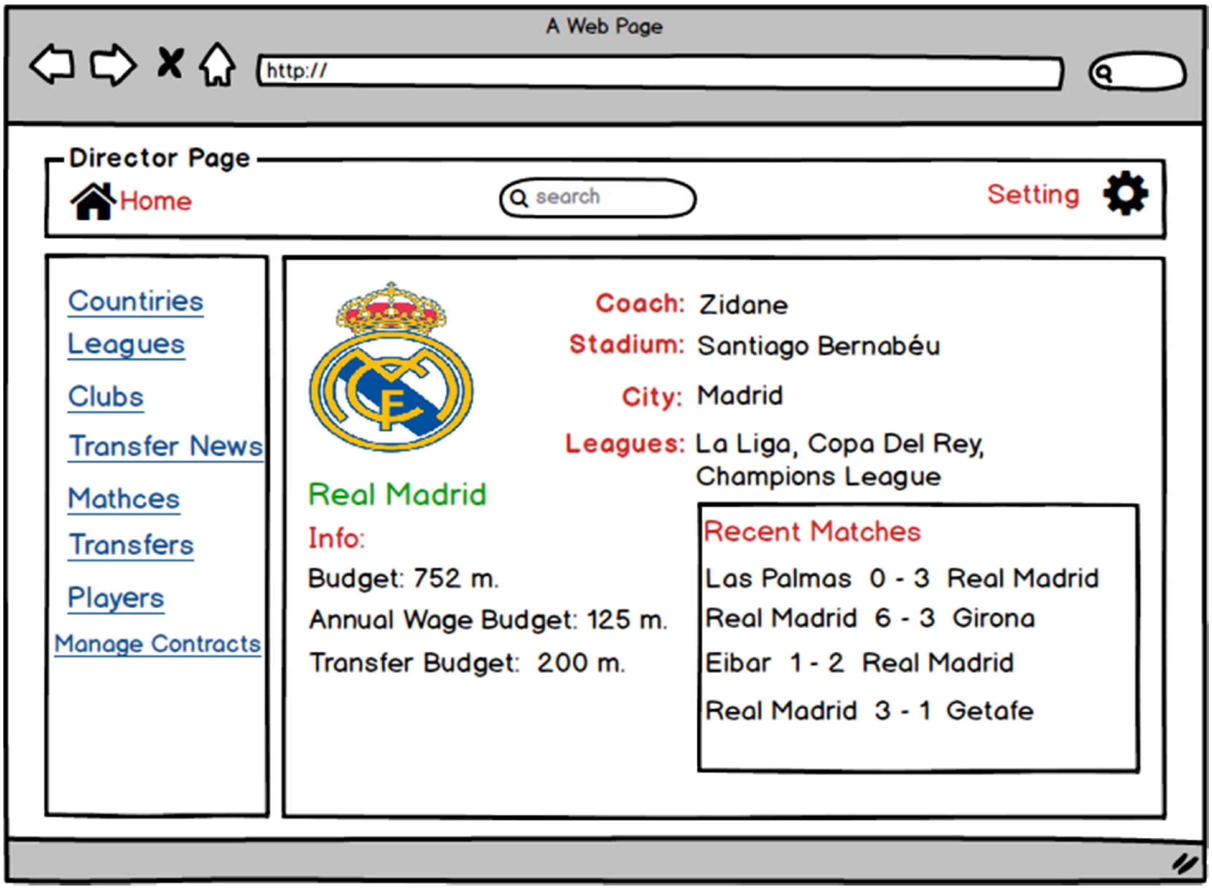
Retrieving Players Info:

```
SELECT  Staff.name, Staff.surname, Staff.salary, Player.value
FROM Player, Staff
WHERE Player.ID = Staff.ID
```

Searching in Players:

```
SELECT Staff.name, Staff.surname, Staff.salary, Player.value
FROM Player, Staff
WHERE Player.ID = Staff.ID AND (Staff.name = @searchText OR Staff.surname =
@searchText)
```

5.7 Directors Page



Coach: Zidane

Stadium: Santiago Bernabéu

City: Madrid

Leagues: La Liga, Copa Del Rey, Champions League

Real Madrid

Info:

Budget: 752 m.

Annual Wage Budget: 125 m.

Transfer Budget: 200 m.

Recent Matches

Las Palmas 0 - 3 Real Madrid

Real Madrid 6 - 3 Girona

Eibar 1 - 2 Real Madrid

Real Madrid 3 - 1 Getafe

Input: @searchText

Process: Directors page visualizes informations about the club that is directed by current user. Informations of the directed club are directed club’s club symbol, budget, annual wage budget transfer budget, coach, stadium, city, league and recent matches. Moreover, in left part of the page in addition to guests page there is a additional link to send user to a page that director can menage contracts of his players and coach. Top of the page is the same as users page. In order to enter this page you should an director.

SQL Statements:

Retrieving most of the information is same with Fan Page.

Retrieving only Director Info:

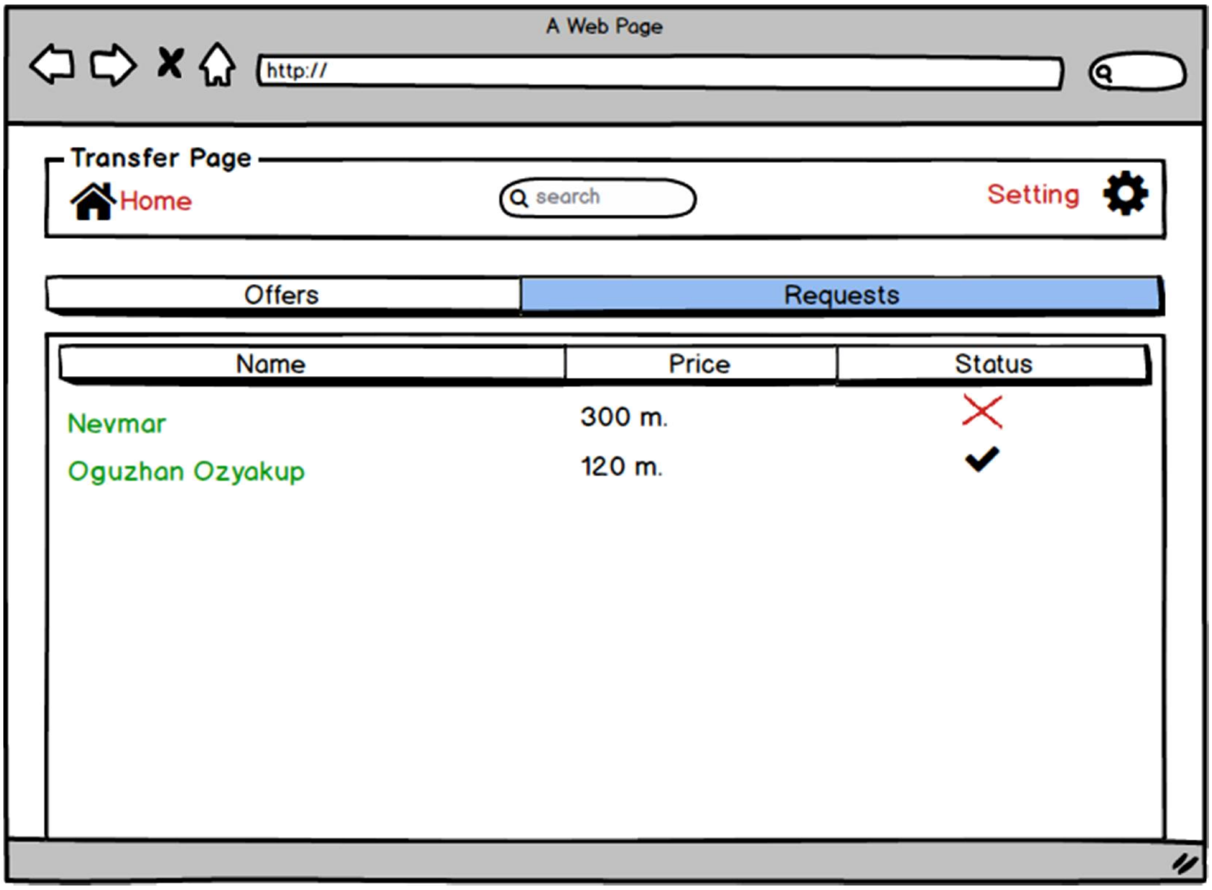
```
SELECT Club.annual_wage_budget, Club.transfer_budget

FROM Director, Club

WHERE Director.ClubID = Club.ID
```

The Search in this page will lead to general Search written in Section 5.3

5.8 Transfer Request Page



Name	Price	Status
Nevmar	300 m.	✗
Oguzhan Ozyakup	120 m.	✓

Input: @searchText, @directorID

Process: In this page director can see the current status of his/her transfer requests status that has been made so far. In this page the name of the player that director made transfer request for, the offered price and the transfers status are visualized. Moreover by clicking offers button director can go to transfer offer page. The top side of the page is not different than directors page's top side. In order to access this page you should be an director.

SQL Statements:

Retrieving Transfer Request:

SELECT Staff.name, Staff.surname, Transfer_Offer.price, Transfer_Offer.status

FROM Transfer_Offer, Player, Staff

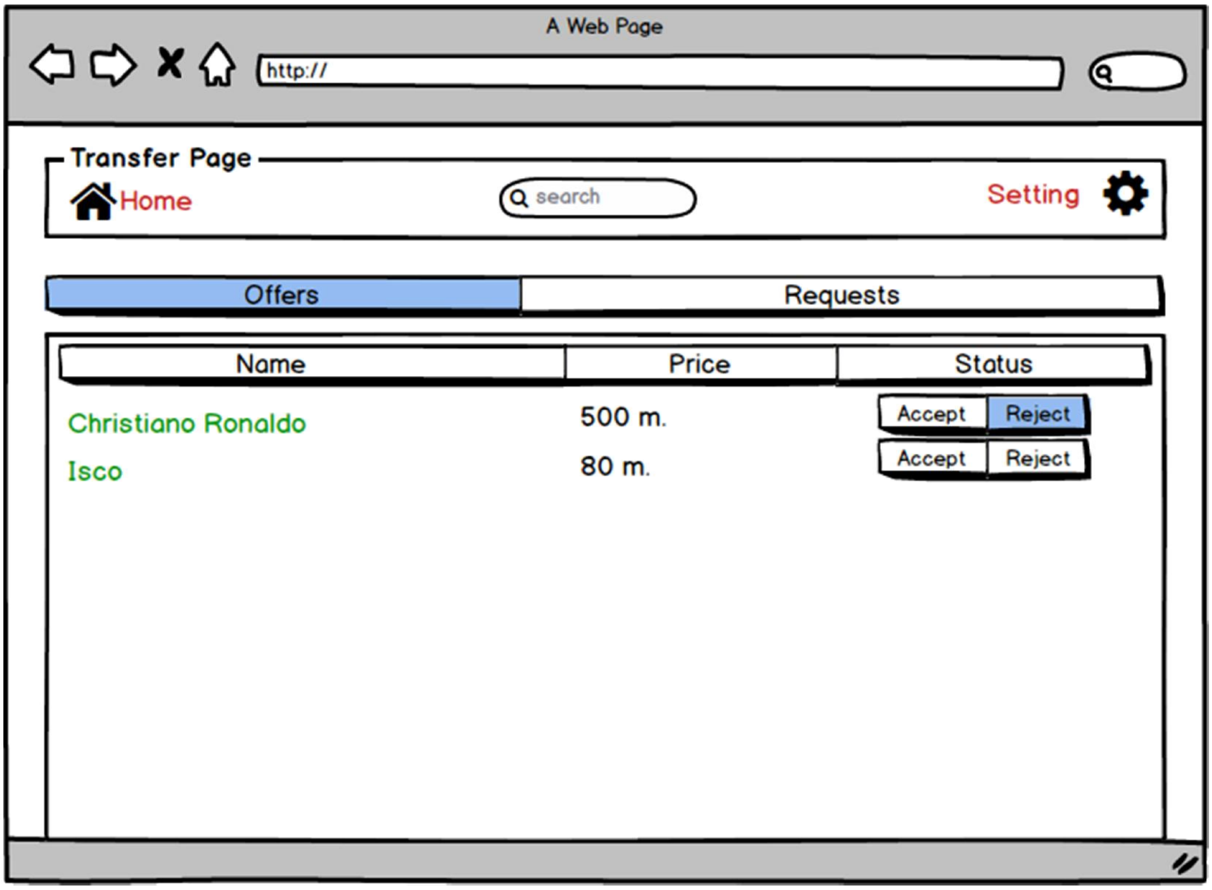
WHERE Transfer_Offer.FromDirectorID = @directorID AND Transfer_Offer.PlayerID = Player.ID AND Player.ID = Staff.ID

Searching in Transfer Requests:

SELECT Staff.name, Staff.surname, Transfer_Offer.price, Transfer_Offer.status

```
        FROM Transfer_Offer, Player, Staff
        WHERE Transfer_Offer.FromDirectorID = @directorID AND Transfer_Offer.PlayerID
= Player.ID AND Player.ID = Staff.ID AND (Staff.name = @searchText OR Staff.surname =
@searchText)
```

5.9 Transfer Offer Page



Name	Price	Status
Christiano Ronaldo	500 m.	Accept Reject
Isco	80 m.	Accept Reject

Input: @searchText

Process: In this page director can see the transfer offerings that is made for the player that he plays in his club. In this page player who get transfer offers name offered price is visualized and in status part there are two buttons in which director can click and accept the offer or reject it. Moreover by clicking offers button director can go to transfer offer page. The top side of the page is not different than directors page's top side. In order to access this page you should be an director.

SQL Statements:

Retrieving Transfer Offer:

SELECT Staff.name, Staff.surname, Transfer_Offer.price, Transfer_Offer.status

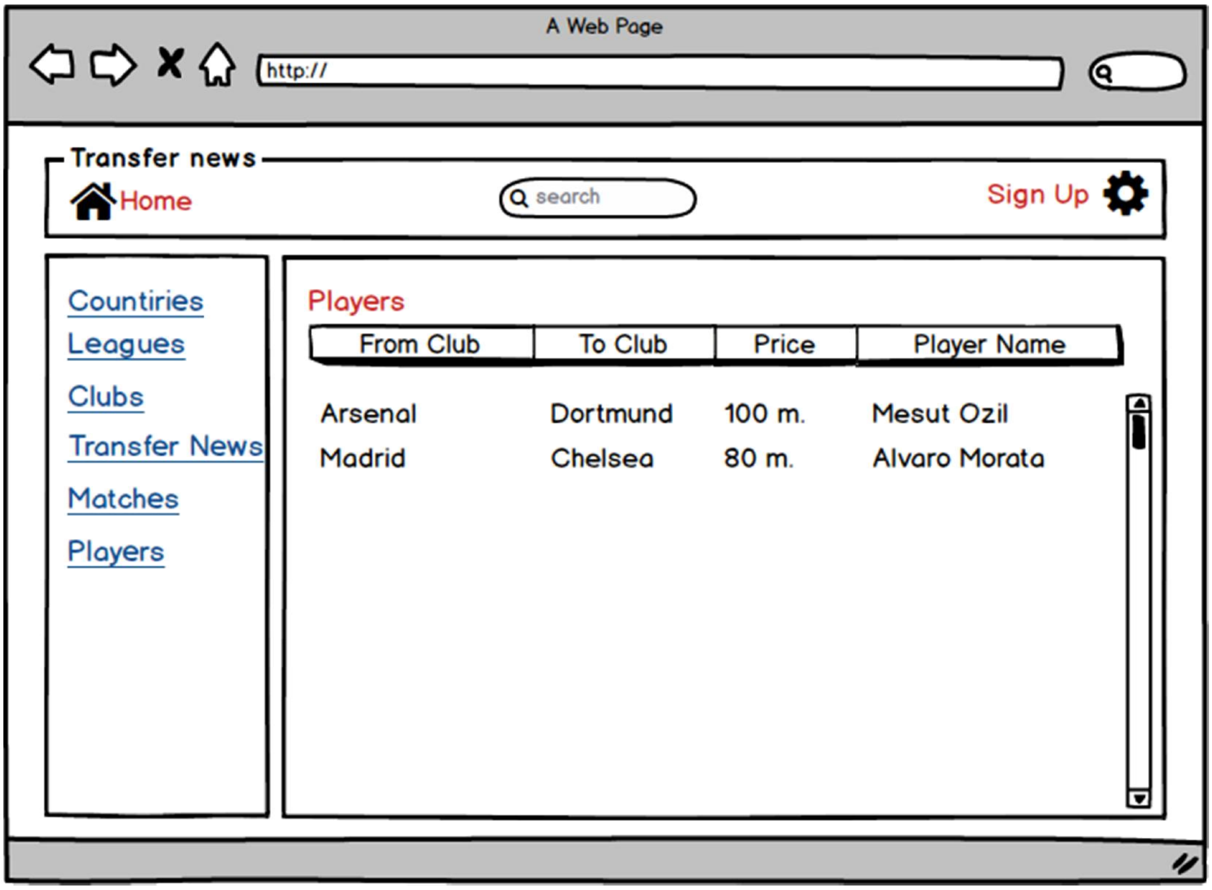
FROM Transfer_Offer, Player, Staff

WHERE Transfer_Offer.ToDirectorID = @directorID AND Transfer_Offer.PlayerID = Player.ID AND Player.ID = Staff.ID

Searching in Transfer Offers:

```sql
SELECT Staff.name, Staff.surname, Transfer_Offer.price, Transfer_Offer.status
FROM Transfer_Offer, Player, Staff
WHERE Transfer_Offer.ToDirectorID = @directorID AND Transfer_Offer.PlayerID = Player.ID AND Player.ID = Staff.ID AND (Staff.name = @searchText OR Staff.surname = @searchText)
```

5.10 Transfer News Page



From Club	To Club	Price	Player Name
Arsenal	Dortmund	100 m.	Mesut Ozil
Madrid	Chelsea	80 m.	Alvaro Morata

Input: @searchText

Process: In this page the transfers are listed. In this page player that was transfered's previous club, current club, transfer price and name is visualized. The left and top of the page will be the same as guest's page. This page is accessible by everybody.

SQL Statements:

Retrieving Transfer News:

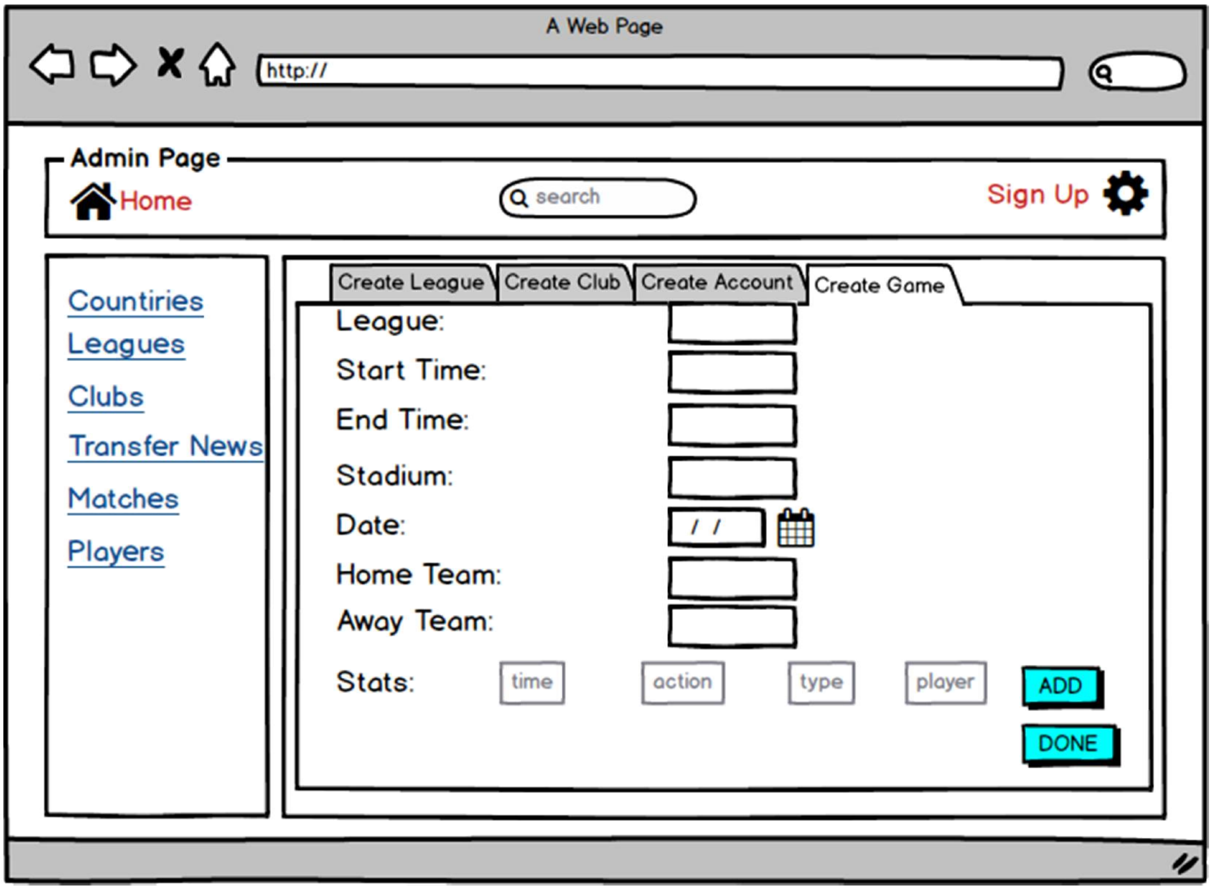
```
with firstClubNames(offerID, name) as(
SELECT TO.ID, Club.name
FROM Transfer_Offer TO, Club, Director
WHERE TO.status = ‘accepted’ AND TO.FromDirectorID = Director.ID AND DirectorID = Club.ID
)
with secondClubNames(offerID, name) as(
SELECT TO.ID, Club.name
FROM Transfer_Offer TO, Club, Director
WHERE TO.status = ‘accepted’ AND TO.ToDirectorID = Director.ID AND DirectorID = Club.ID
)
with playerInfo(offerID, name, surname, price)(
SELECT TO.offerID, Staff.name, Staff.surname, Player.value
FROM Transfer_Offer TO, Player, Staff
WHERE TO.status = ‘accepted’ AND TO.PlayerID = Player.ID AND Player.ID = Staff.ID
)
SELECT firstClubNames.name, secondClubNames.name, playerInfo.price, playerInfo.name, playerInfo.surname
FROM firstClubNames, secondClubNames, playerInfo
WHERE firstClubNames.offerID = secondClubNames.offerID AND secondClubNames.offerID = playerInfo.offerID
```

Searching for Transfer News:

```
with firstClubNames(offerID, name) as(
SELECT TO.ID, Club.name
FROM Transfer_Offer TO, Club, Director
WHERE TO.status = ‘accepted’ AND TO.FromDirectorID = Director.ID AND DirectorID = Club.ID
)
with secondClubNames(offerID, name) as(
SELECT TO.ID, Club.name
FROM Transfer_Offer TO, Club, Director
WHERE TO.status = ‘accepted’ AND TO.ToDirectorID = Director.ID AND DirectorID = Club.ID
```

```
        )
        with playerInfo(offerID, name, surname, price)(
        SELECT TO.offerID, Staff.name, Staff.surname, Player.value
        FROM Transfer_Offer TO, Player, Staff
        WHERE TO.status = ‘accepted’ AND TO.PlayerID = Player.ID AND Player.ID =
Staff.ID
        )
        SELECT firstClubNames.name, secondClubNames.name, playerInfo.price,
playerInfo.name, playerInfo.surname
        FROM firstClubNames, secondClubNames, playerInfo
        WHERE firstClubNames.offerID = secondClubNames.offerID AND
secondClubNames.offerID = playerInfo.offerID AND (firstClubNames.name = @searchText
OR secondClubNames.name = @searchText OR playersInfo.name = @searchText OR
playersInfo.surname = @searchText)
```

5.11 Admin Create Game Page



Input: @startTime, @endTime, @stadium, @date, @homeTeamID, @awayTeamID, @time, @action, @type, @playerID, @gameID, @leagueID

Process: In this admin can enter a game to football database system by entering its start time, end time, stadium, date home team, away team and the status. The left and top of the page will be the same as guest. In order to access this page the logged account should be an admin account.

SQL Statements:

Inserting into Game:

INSERT into Game

values(@gameID, @startTime, @endTime, @stadium, @date, @homeTeamID, @awayTeamID);

Inserting into Stats:

INSERT into Stats

values(@time, @action, @type, @gameID, @playerID);

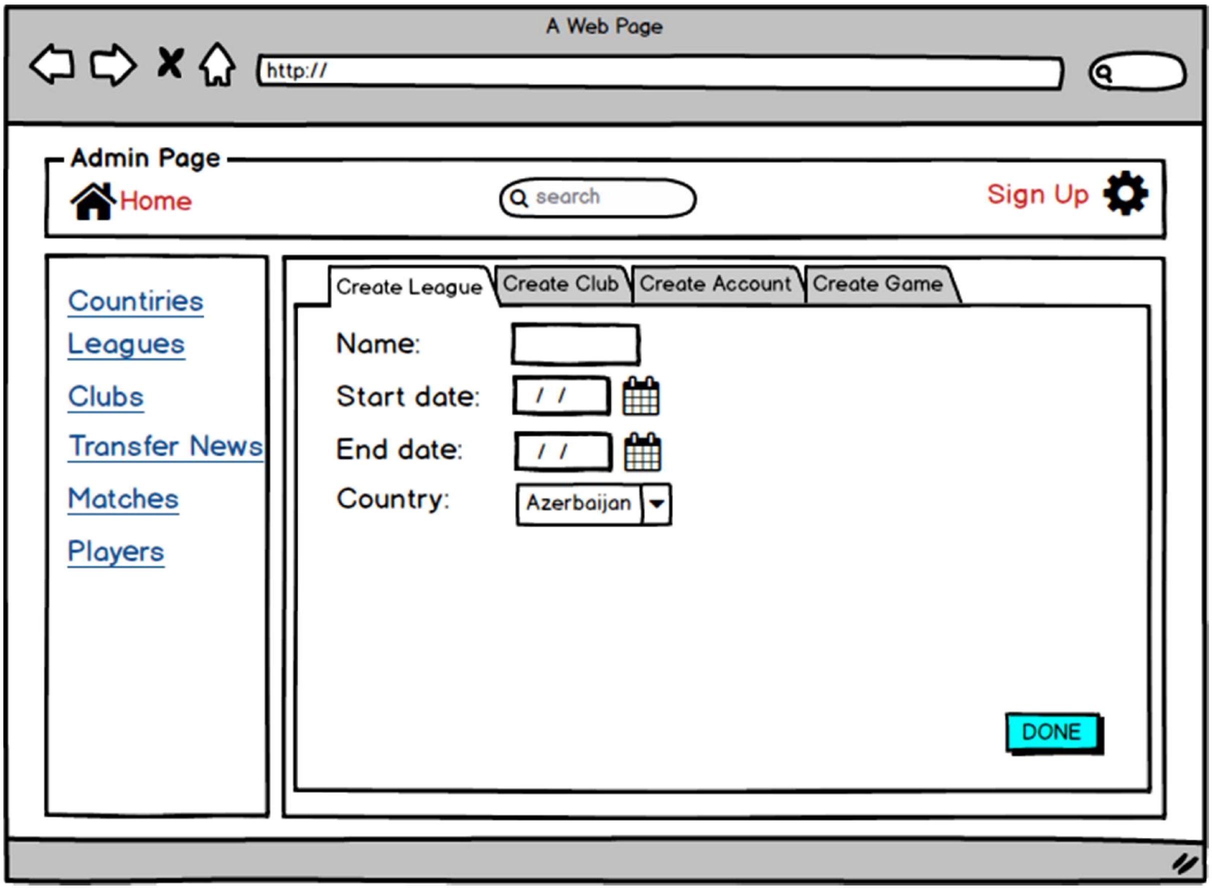
Inserting into League_Game:

```
INSERT into League_Game
values(@leagueID, @gameID);
```

Inserting into Player_Game:

```
INSERT into Player_Game:
values(@playerID, @gameID);
```

5.12 Admin Create League Page



Input: @name, @startDate, @endDate, @country, @leagueID

Process: In this admin can enter a league to football database system by entering its start name, start date, end date and country. The left and top of the page will be the same as guest. In order to access this page the logged account should be an admin account.

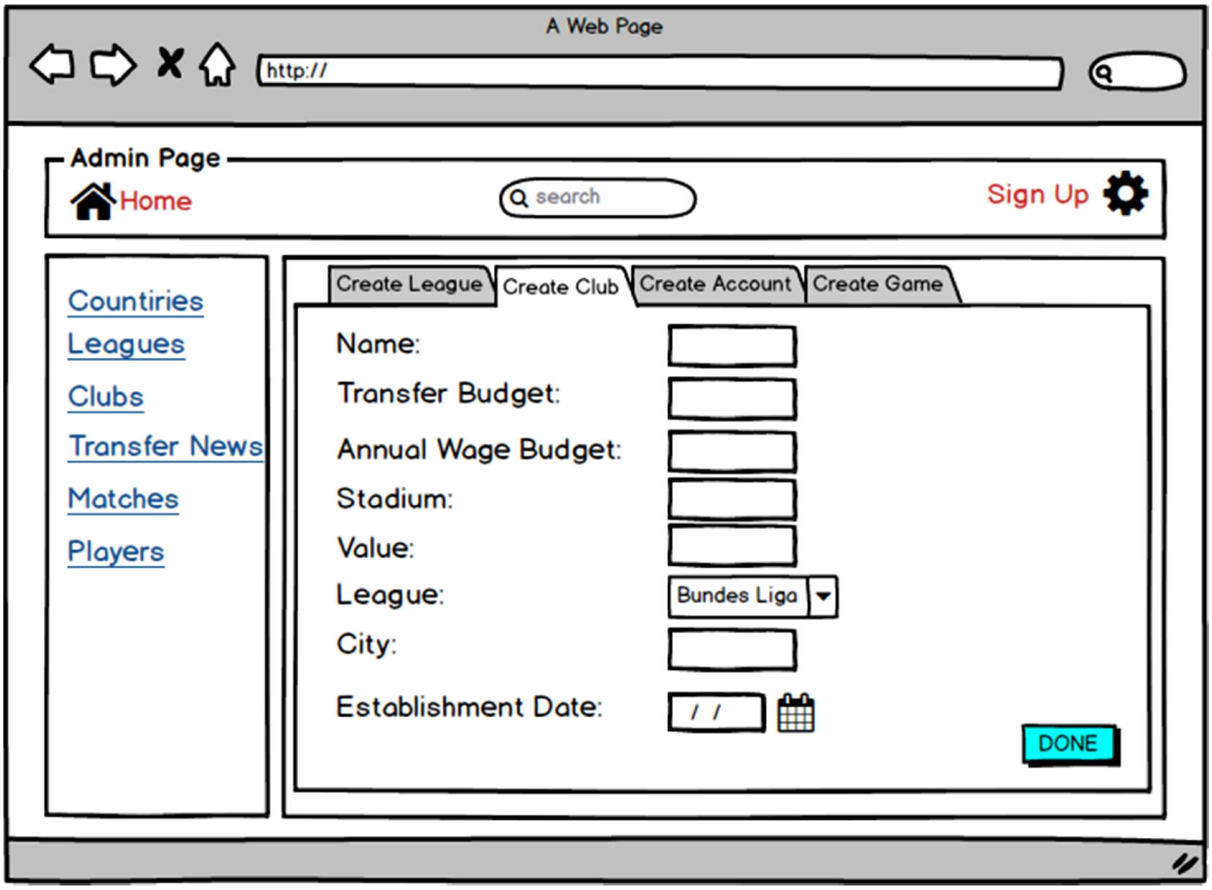
SQL Statements:

Inserting into League:

INSERT into LEAGUE

values(@league, ‘not determined’, @name, @startDate, @endDate, @country);

5.13 Admin Create Club Page



Name:

Transfer Budget:

Annual Wage Budget:

Stadium:

Value:

League: Bundes Liga

City:

Establishment Date: / /

Input: @name, @transferBudget, @annualWageBudget, @leagueID, @city, @establishmentDate, @clubID, @stadium, @value

Process: In this admin can enter a club to football database system by entering its start name, transfer budget, annual wage budget, league, city and establishment date. The left and top of the page will be the same as guest. In order to access this page the logged account should be an admin account.

SQL Statements:

Inserting Into Clubs:

INSERT into Club

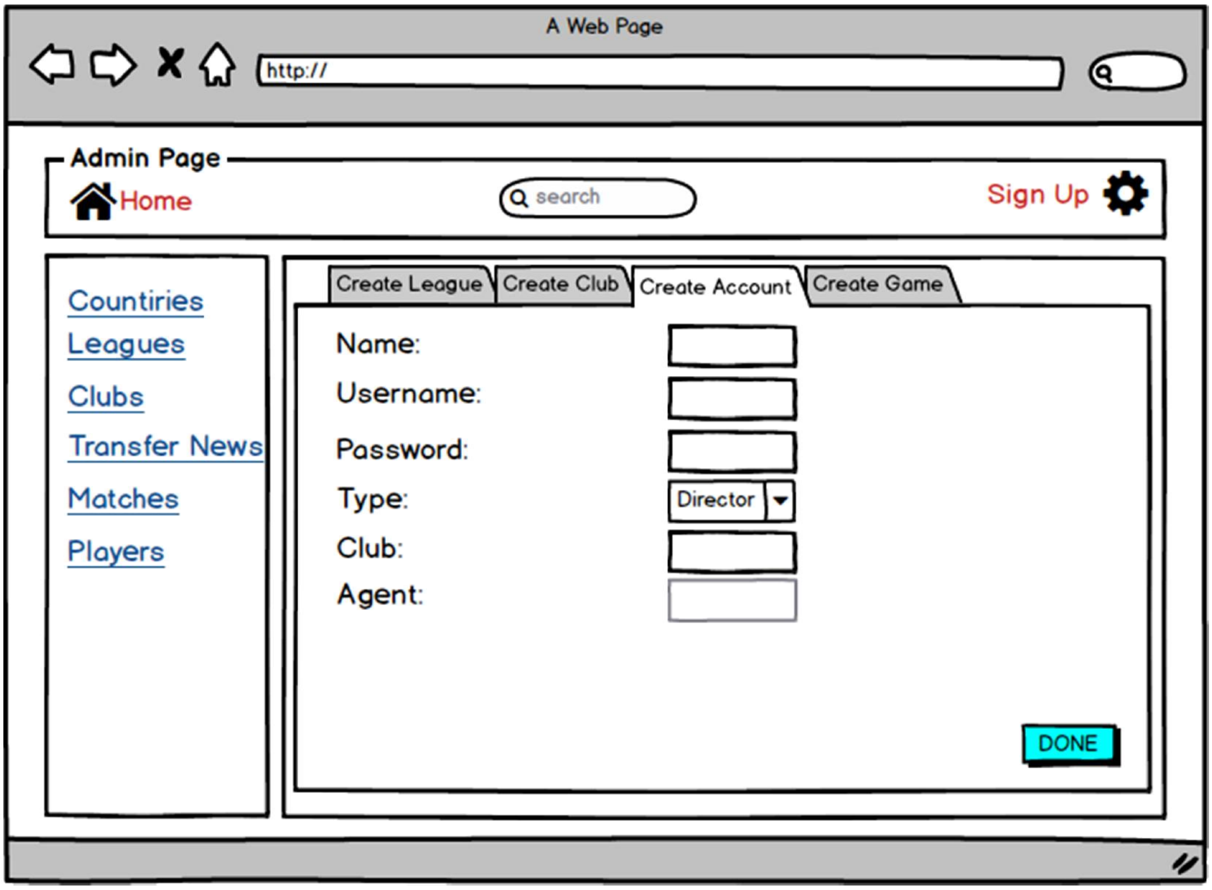
values(@clubID, @name, @transferBudget, @annualWageBudget, @city, @establishmentDate, @value, @stadium);

Inserting Into League_Club:

INSERT into League_Club:

```
        values(@leagueID, @clubID);
```

5.14 Admin Create Account Page



Input: @username,@clubID,@playerID,@type,@agentID, @name,@age,@salary,@nationality,@value, @player_position,@player_height, @surname,@birthdate

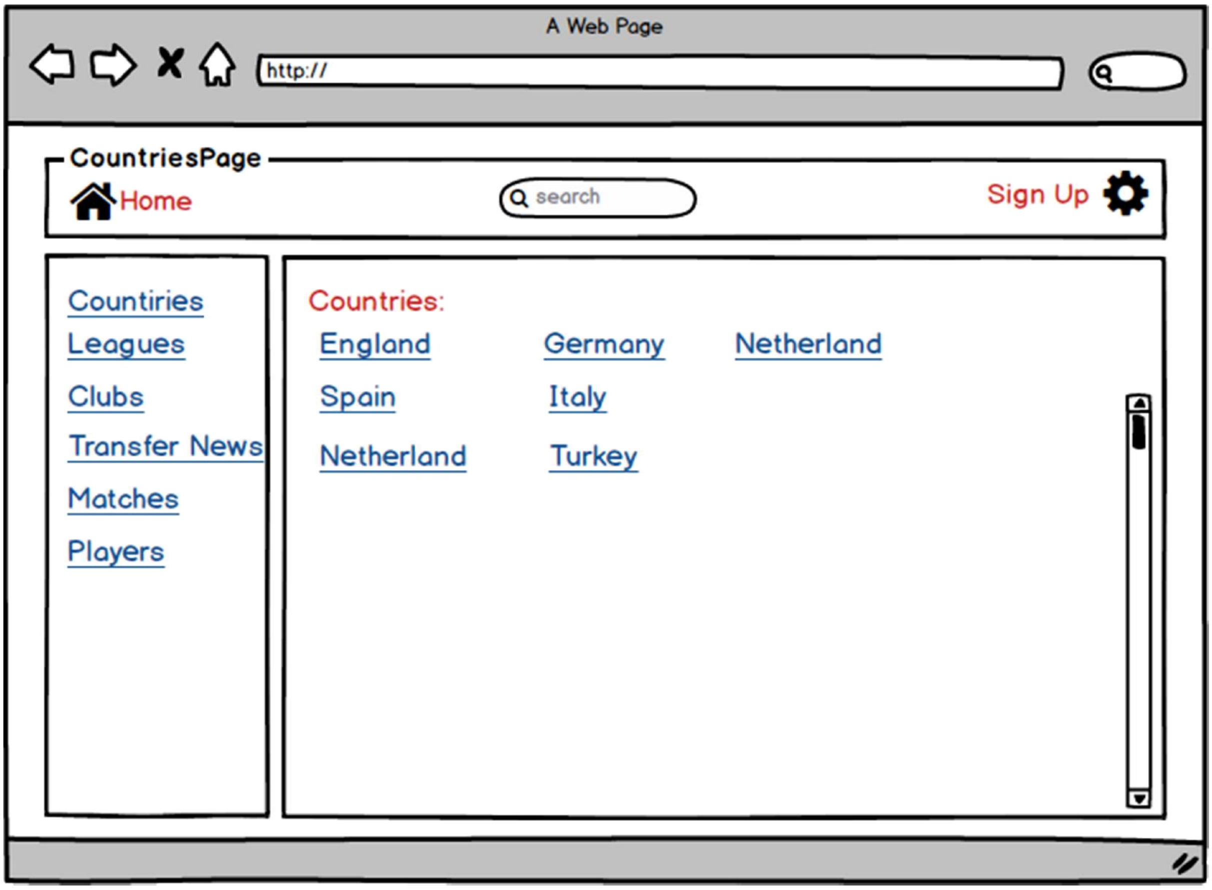
Process: In this admin can enter a account to football database system by entering its start name, username, password, type, club and agent if the type is player or coach. Therefore, this is the place that new directors, players, agents and coaches are added to the database system. The left and top of the page will be the same as guest. In order to access this page the logged account should be an admin account.

SQL Statements:

```
    INSERT INTO user ( username, password, type)
    VALUES( @username,@password,@type )
        IF @type IS “Director”
          BEGIN
              INSERT INTO Director( DirectorID, ClubID )
```

```
            VALUES(@directorID, @clubID)
          INSERT INTO Staff ( ID,name,surname, age(), salary, nationality, birthdate )
            VALUES(
@directorID,@name,@surname,@age,@salary,@nationality,@birthdate)
        END
    ELSE IF @type IS “Agent”
        BEGIN
             INSERT INTO Agent(AgentID)
               VALUES  (@agentID)
            INSERT INTO Staff ( ID,name,surname, age(), salary, nationality, birthdate )
              VALUES(
@agentID,@name,@surname,@age,@salary,@nationality,@birthdate)
        END
   ELSE IF @type IS “Player”
        BEGIN
            INSERT INTO Player( ID,position,value,height, AgentID )
              VALUES( @playerID, @player_position,@value,@player_height, @agentID
)
            INSERT INTO Plays(PlayerID, ClubID)
              VALUES(@playerID,@clubID)
            INSERT INTO Staff ( ID,name,surname, age(), salary, nationality, birthdate )
              VALUES( @playerID,@name, @surname, @age, @salary, @nationality,
@birthdate)
        END
   ELSE
        BEGIN
            INSERT INTO Coach(ID, value,  AgentID, ClubID)
              VALUES (@coachID,@value, @agentID,@clubID)
            INSERT INTO Staff ( ID, name,surname, age(), salary, nationality, birthdate )
              VALUES(
@coachID,@name,@surname,@age,@salary,@nationality,@birthdate )
      END
```

5.15 Countries Page



Input: @searchText

Process: In this page the countries that is added to football database system are listed. If the link is clicked it will sent us to the country's league page for this country. The left and top of the page will be the same as guest. This page is accessible for every user.

SQL Statements:

Listing Countries:

```
SELECT Country.name

FROM Country
```

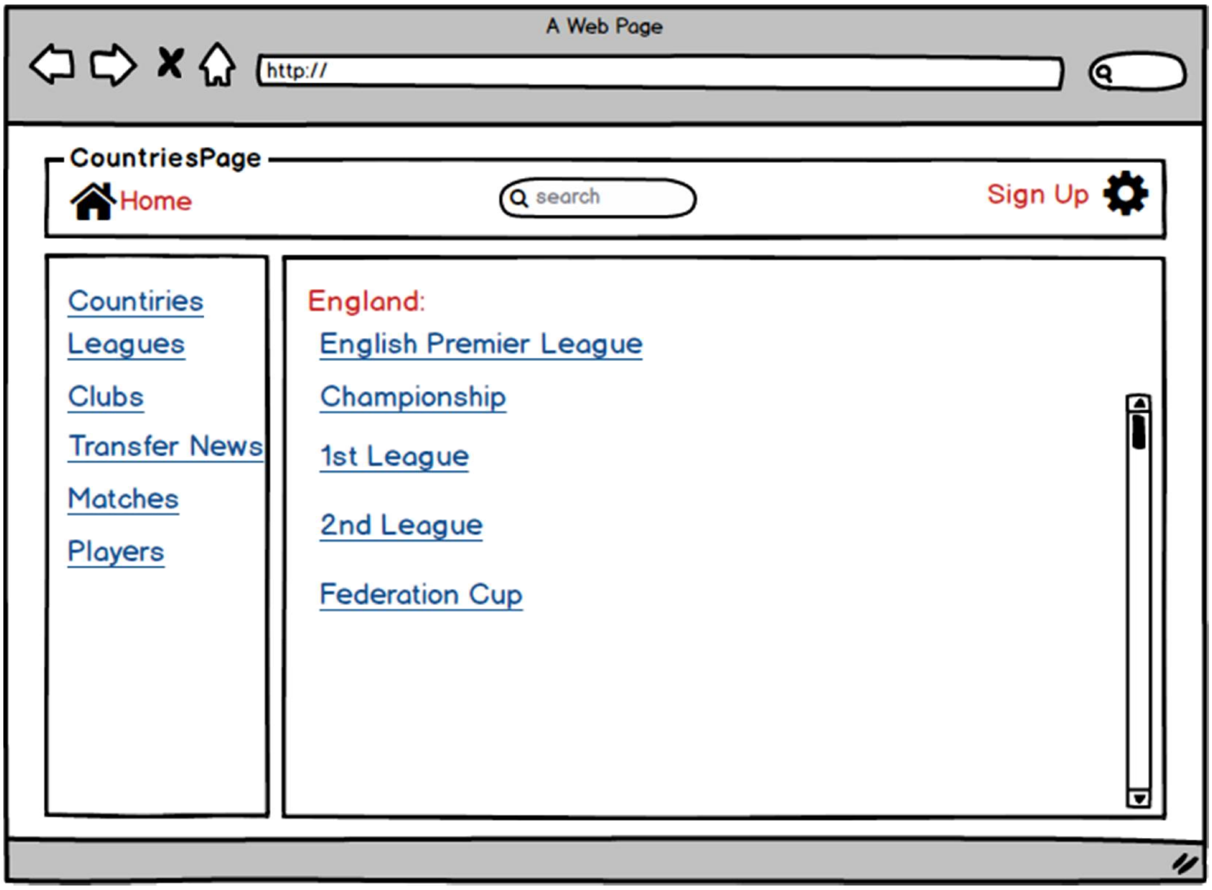
Searching For a Country:

```
SELECT Country.name

FROM Country

WHERE Country.name = @searchText
```

5.16 Country's Leagues Page



Input: @searchText, @country

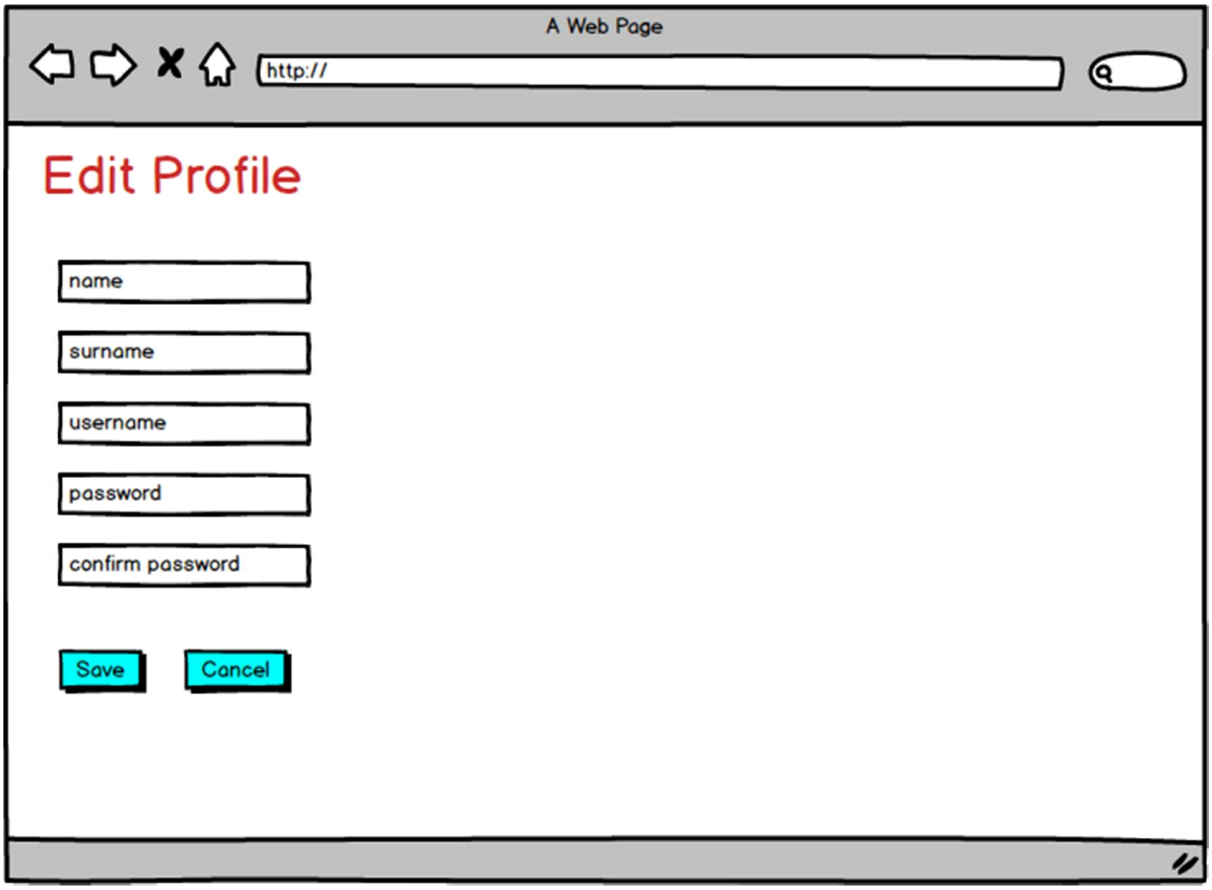
Process: In this page the a county's leagues are displayed. If the link is clicked it will sent us to the league page of chosen league. The left and top of the page will be the same as guest. This page is accessible for every user.

SQL Statements:

Listing Leagues in a Country:

```
SELECT League.name
FROM League
WHERE League.CountryName = @country
```

5.17 Edit Profile Page



Input:

@input_name,@input_surname,@input_username,@input_password,@input_confirm_password

Process: In this page user can edit his/her profile. There are text box that user should type in in order to edit his/her profile and the system ask to enter name, surname, password, username and confirmed password. In order to access this page the user should have an account.

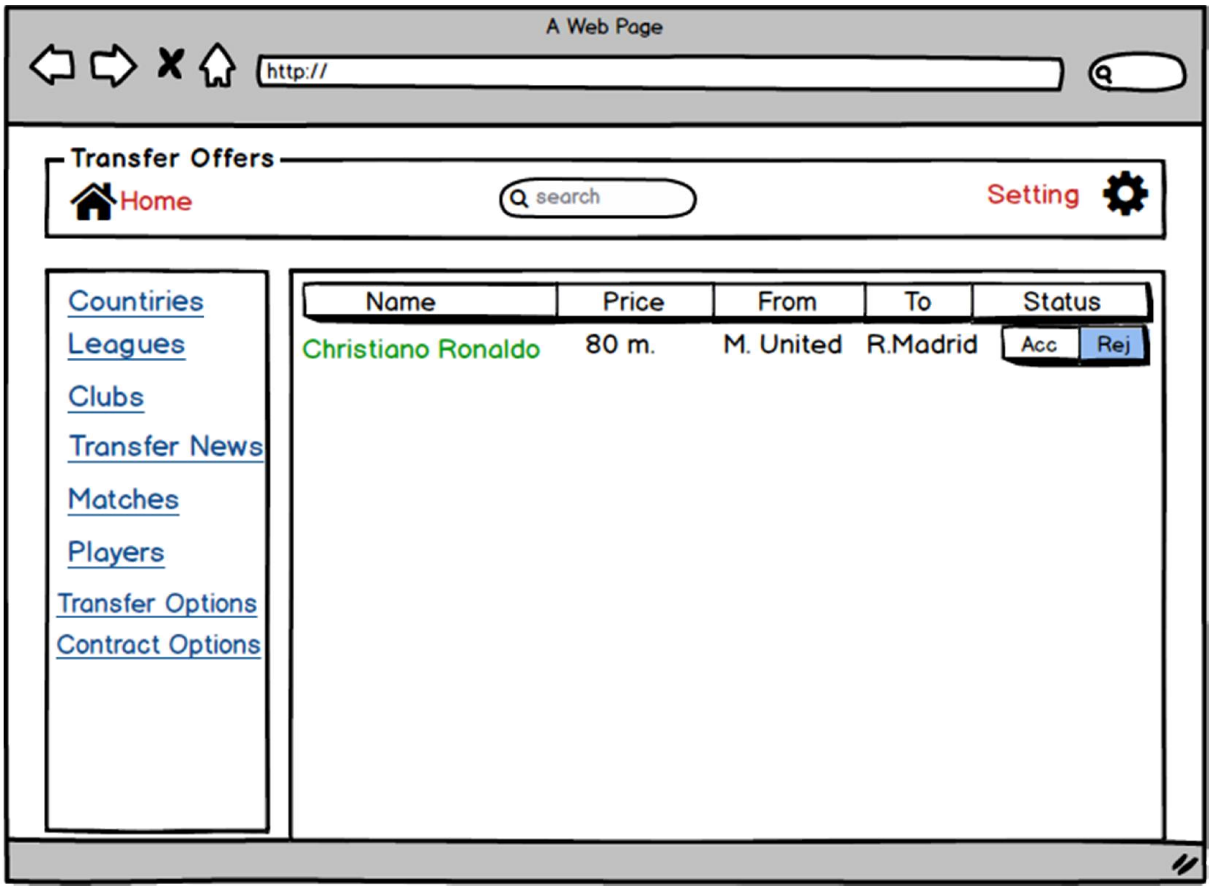
SQL Statements:

UPDATE user

SET name = @input_name, surname = @input_surname, username = @input_username, password = @input_password

WHERE username = @username

5.18 Transfer Offers Page



Name	Price	From	To	Status
Christiano Ronaldo	80 m.	M. United	R.Madrid	Acc Rej

Input: @searchText, @agentID

Process: In this page agent will see the transfer offer that is accepted by the club director. This is the final part of the transfer if the agent will accept the transfer offer player's transfer will be happened. In the system players name, offered price current club and offering club is listed by clicking accept and reject button agent can make the decision. The left and top of the page will be the same as guest other than transfer options and contract options. This page is accessible for only agents.

SQL Statements:

Retrieving Transfer Options:

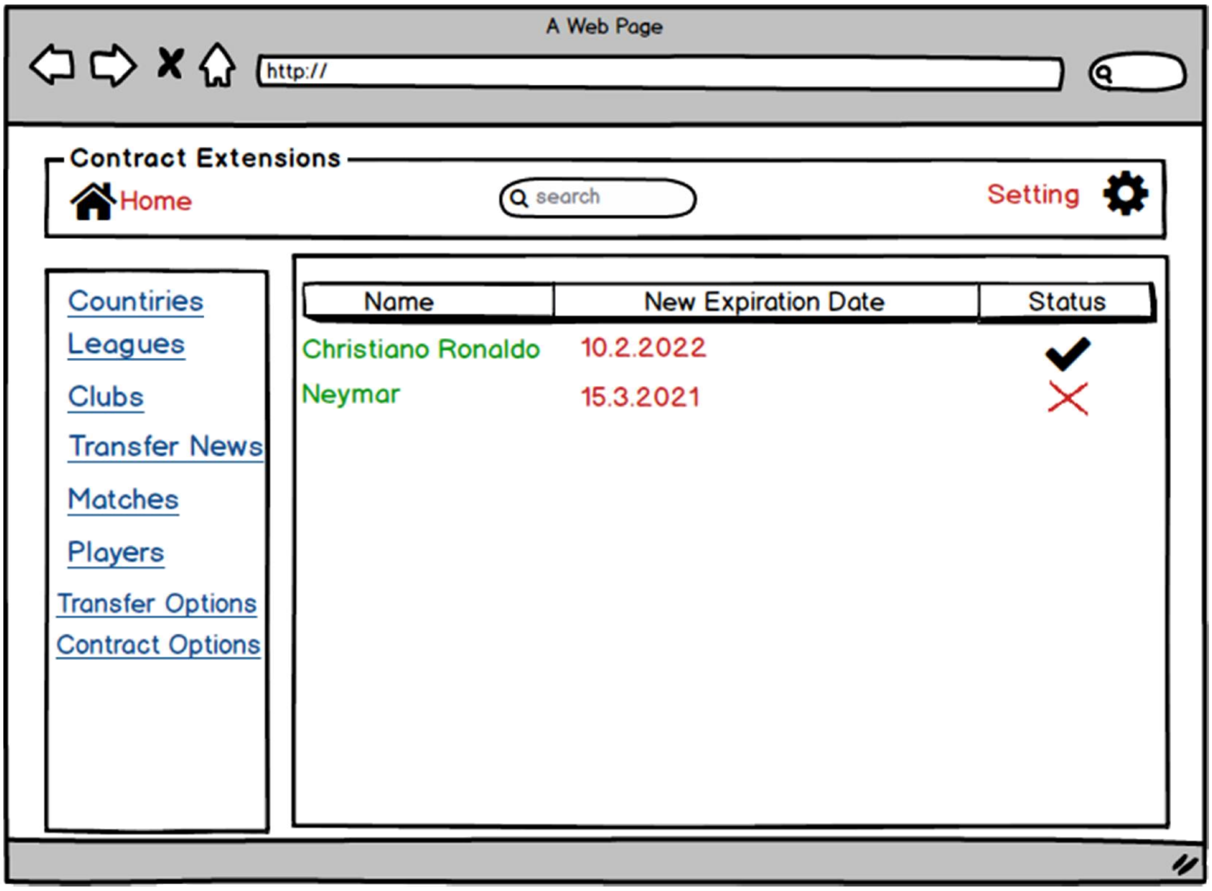
```
with firstClubNames(offerID, name) as(
SELECT TO.ID, Club.name
FROM Transfer_Offer TO, Club, Director
WHERE TO.status = ‘accepted’ AND TO.FromDirectorID = Director.ID AND DirectorID = Club.ID
)
with secondClubNames(offerID, name) as(
SELECT TO.ID, Club.name
FROM Transfer_Offer TO, Club, Director
WHERE TO.status = ‘accepted’ AND TO.ToDirectorID = Director.ID AND DirectorID = Club.ID
)
with playerInfo(offerID, name, surname, price, status)(
SELECT TO.offerID, Staff.name, Staff.surname, Player.value, TO.status
FROM Transfer_Offer TO, Player, Staff
WHERE TO.status = ‘accepted’ AND TO.PlayerID = Player.ID AND Player.ID = Staff.ID AND Player.AgentID = @agentID
)
SELECT firstClubNames.name, secondClubNames.name, playerInfo.price, playerInfo.name, playerInfo.surname, playerInfo.status
FROM firstClubNames, secondClubNames, playerInfo
WHERE firstClubNames.offerID = secondClubNames.offerID AND secondClubNames.offerID = playerInfo.offerID
```

Searching for Transfer Options:

```
with firstClubNames(offerID, name) as(
SELECT TO.ID, Club.name
FROM Transfer_Offer TO, Club, Director
WHERE TO.status = ‘accepted’ AND TO.FromDirectorID = Director.ID AND DirectorID = Club.ID
)
```

```
with secondClubNames(offerID, name) as(
SELECT TO.ID, Club.name
FROM Transfer_Offer TO, Club, Director
WHERE TO.status = ‘accepted’ AND TO.ToDirectorID = Director.ID AND DirectorID = Club.ID
)
with playerInfo(offerID, name, surname, price, status)(
SELECT TO.offerID, Staff.name, Staff.surname, Player.value, TO.status
FROM Transfer_Offer TO, Player, Staff
WHERE TO.status = ‘accepted’ AND TO.PlayerID = Player.ID AND Player.ID = Staff.ID AND Player.AgentID = @agentID
)
SELECT firstClubNames.name, secondClubNames.name, playerInfo.price, playerInfo.name, playerInfo.surname, playerInfo.status
FROM firstClubNames, secondClubNames, playerInfo
WHERE firstClubNames.offerID = secondClubNames.offerID AND secondClubNames.offerID = playerInfo.offerID AND (firstClubNames.name = @searchText OR secondClubNames.name = @searchText OR playersInfo.name = @searchText OR playersInfo.surname = @searchText)
```

5.19 Contract Extensions Page



Name	New Expiration Date	Status
Christiano Ronaldo	10.2.2022	✓
Neymar	15.3.2021	✗

Input: @searchText, @agentID

Process: In this page agent will see a list of players who he works with and in this list player's name, new expiration date and status which shows whether director wants to extend the contract or not. The left and top of the page will be the same as guest other than transfer options and contract options. This page is accessible for only agents.

SQL Statements:

Showing Agent's Players Information:

SELECT Player.name, Contract.expiration_date, Contract.status

From Player, Contract

Where Player.AgentID = @agentID AND Contract.AgentID = @agentID AND Contract.status = "pending"

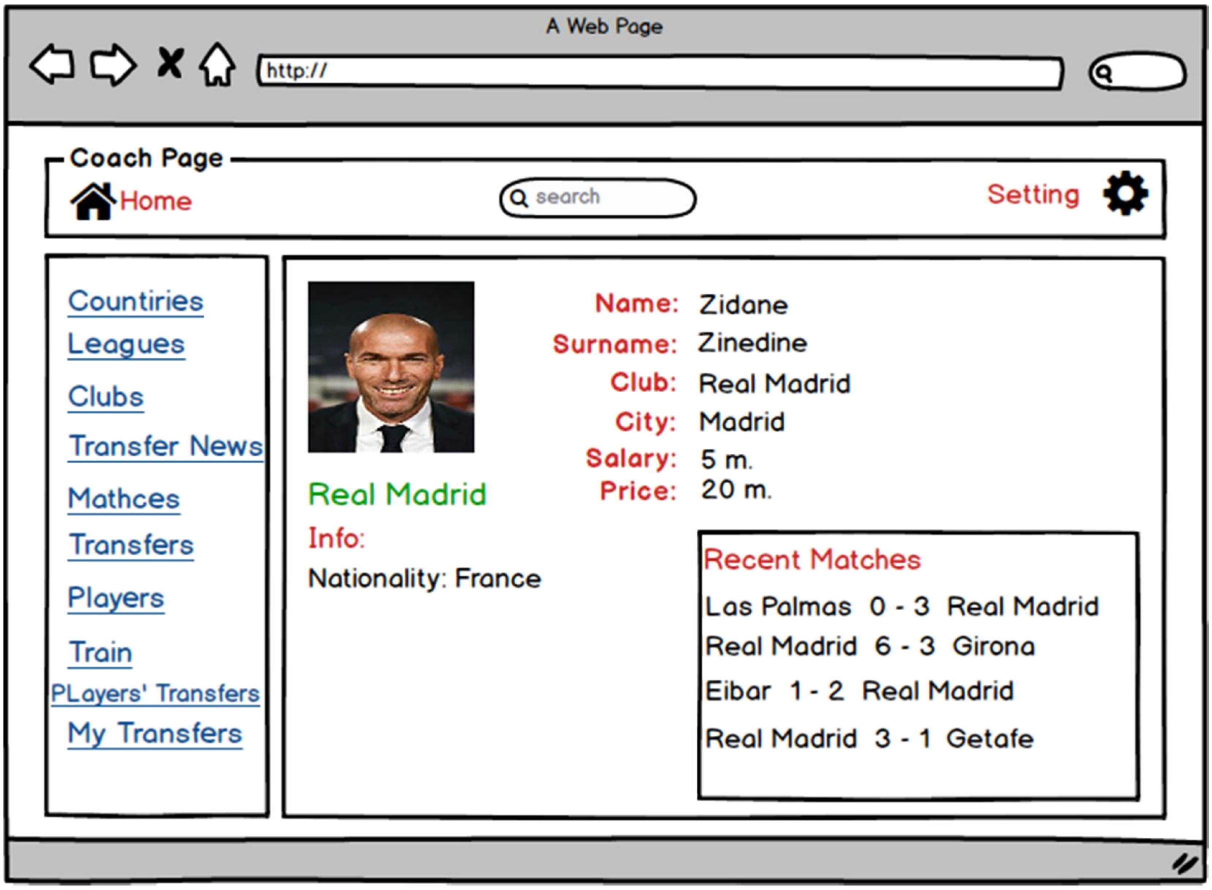
Search

```
SELECT Player.name, Contract.expiration_date, Contract.status

From Player, Contract

Where Player.AgentID = @agentID AND  Contract.AgentID = @agentID Contract.status = “pending” AND Player.name = @searchText
```

5.20 Coach Page



Name: Zidane
Surname: Zinedine
Club: Real Madrid
City: Madrid
Salary: 5 m.
Price: 20 m.

Real Madrid
Info:
Nationality: France

Recent Matches
Las Palmas 0 - 3 Real Madrid
Real Madrid 6 - 3 Girona
Eibar 1 - 2 Real Madrid
Real Madrid 3 - 1 Getafe

Input: @searchText, @coachID

Process: In this page coach can see his own profile. In the page coach's nationality, name, surname, club city sallary price and his clubs last matches. The left and top of the page will be the same as guest other than train players' transfers and my transfers links. My transfer link sends coach to a link that gives information about the about the transfer status of him. This page is accessible for only coaches but this coach informations is not only visible for coaches but also other users.

SQL Statements:

Coaches infos

SELECT Staff.nationality, Staff.name, Staff.surname, Staff.salary, Coach.value, Club.name

FROM Coach,Club, Staff

WHERE Staff.ID = @coachID AND @coachID = coach.ID AND coach.ClubID = Club.ID

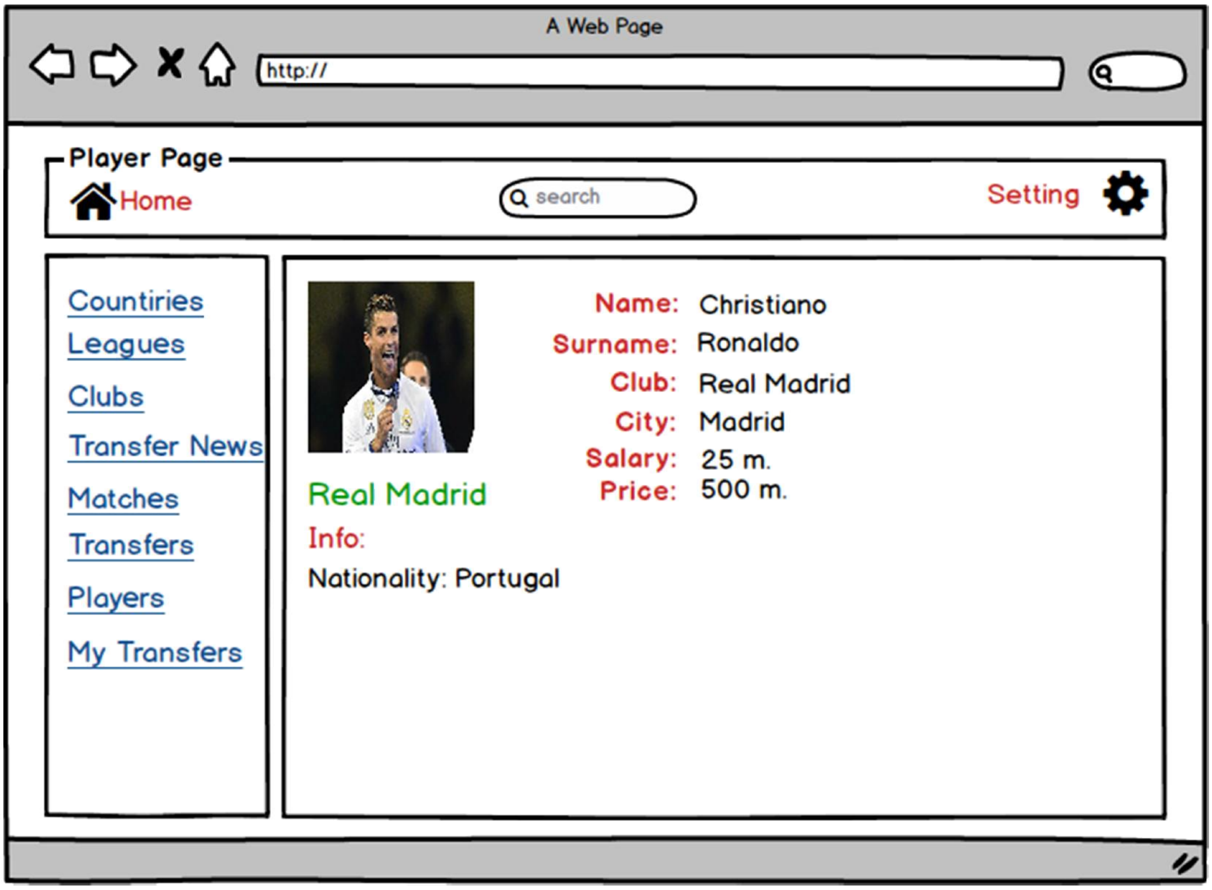
Recent Matches

It has already been implemented in fan page.

Search

Search will be general search.

5.21 Player Page



Name: Christiano
Surname: Ronaldo
Club: Real Madrid
City: Madrid
Salary: 25 m.
Price: 500 m.
Nationality: Portugal

Input: @searchText, @playerID

Process: In this page player can see his own profile. In the page players's nationality, name, surname, club, city, salary and price. The left and top of the page will be the same as guest other than train players' transfers and my transfers links. My transfer link sends player to a link that gives information about the about the transfer status of him. This page is accessible for only players but this informations are not only visible for only football players but also other users.

SQL Statements:

Coaches infos

SELECT Staff.nationality, Staff.name, Staff.surname, Staff.salary, Player.value, Club.name, Club.city

FROM Player,Club, Staff

WHERE Staff.ID = @playerID AND @playerID = Player.ID AND Player.ClubID = Club.ID

Recent Matches

It has already been implemented in fan page.

Search

Search will be general search.

6.Advanced Database Components

6.1. Views

6.1.1 Ongoing Transfers

The transfers which are currently not fully resolved can be viewed only by participating actors such as a player involved, an agent involved and corresponding directors involved. Therefore, we created this so that relevant users can see ongoing transfers.

```
create view ongoingTransfers as
SELECT TO.ID, TO.price, TO.transferDate, TO.PlayerID, TO.FromDirectorID,
TO.ToDirectorID
FROM Transfer_Offers TO
WHERE TO.status = ‘pending’
```

6.1.2 Ongoing Contract Extension Request

Contract Extensions can only be viewed by participating actors agent, player and director. This view is used in order to show appropriate users to see this data.

```
create view ongoingContracts as
SELECT Contract.PlayerID, Contract.DirectorID, Contract.AgentID, Contract.bonus,
Contract.expiration_date
FROM Contract
WHERE Contract.status = ‘pending’
```

6.1.3 Financial Details

Financial details of a club such as annual wage budget and transfer budget can be viewed only by a director of a club. Thus, we have created this view.

```
 CREATE VIEW financial_details AS
 SELECT Club.ID, Club.annual_wageBudget, Club.transfer_budget
 FROM Club, Director
 WHERE Club.ID = Director.ClubID
```

6.1.4 Favorite Team’s Recent Matches

This view lists all the matches and their scores of the favorite team of a fan.

```
        with favTeam(ID) as (
        SELECT Fan.favoriteTeamID
        FROM Fan
        WHERE Fan.ID = @curUserID
        )
        with games(ID, homeID, awayID, homeTeamName, awayTeamName) as(
        SELECT Game.ID, Game.home_teamID, Game.away_teamID, Club1.name,
        Club2.name
        FROM (SELECT * FROM Game ORDER BY Game.date DESC) G, Club Club1, Club
Club2
        WHERE Game.home_teamID = Club1.ID AND Game.away_teamID = Club2.ID AND
(G.home_teamID = favTeamID OR G.away_teamID = favTeamID)
)
with homeGoals(ID, no) as(
        SELECT games.homeID, count(*)
        FROM games, stats, plays
        WHERE games.ID = stats.GameID AND plays.PlayerID = stats.PlayerID AND
        plays.ClubID = games.homeID AND stats.action = ‘goal’
        )
with awayGoals(ID, no) as(
        SELECT games.awayID, count(*)
        FROM games, stats, plays
        WHERE games.ID = stats.GameID AND plays.PlayerID = stats.PlayerID AND
        plays.ClubID = games.awayID AND stats.action = ‘goal’
        )
```

```
SELECT games.homeTeamName as homeTeamName, games.awayTeamName as
awayTeamName,  homeGoals.no as homeGoals,  awayGoals.no as awayGoals
    FROM games, homeGoals, awayGoals
    WHERE games.homeID = homeGoals.ID AND games.awayID = awayGoals.ID
```

6.2. Stored Procedures

We are going to use procedure for game stats. Since stats of the game will be calculated same for every new match. We will pass the match id to the procedure and then calculate stats for that match. Moreover, we will use stored procedures for general search. We will pass user inputs to this procedure and search for it in database accordingly.

6.3. Reports

6.3.1. Report of total expenses of clubs

```
        WITH player_salary ( playerSalary , clubID  ) AS (
            SELECT Staff.salary as playerSalary, Club.ID as clubID
             FROM   Player, Club, Staff, plays
             WHERE Player.ID = plays.PlayerId AND plays.ClubID = Club.ID
                         AND staff.ID = Player.ID
            )
     WITH coach_salary ( coachSalary , clubID  ) AS (
              SELECT Staff.salary as coachSalary, Club.ID as clubID
              FROM   Coach, Club, Staff
             WHERE Coach.ClubID = Club.ID AND Coach.ID = Staff.ID
            )
     CREATE VIEW total_expenses
```

```
                    AS SELECT  player_salary.playerSalary +
coach_salary.coachSalary     ,player_salary.clubID
      FROM     player_salary, coach_salary
      WHERE  player_salary.clubID = coach_salary.clubID
```

6.4. Triggers

- When director offers transfer for player, it automatically updates the transfer news.
- When transfer offer has completed, financial details of participating buyer and seller clubs will be changed.
- When player changes his team, tables that hold these informations will be changed accordingly.
- When coach changes his team, his attributes will be changed accordingly.
- When new match is inserted to matches, stats of the clubs that participate in the match will change accordingly.

6.5. Constraints

- Only director can see financial details of the club.
- Only fans can have favorite teams and subscribe to clubs.
- Only director can extend the contract of the player.
- Only director can offer transfer for another player and only him can accept/reject received transfers.

7.Implementation Plan

We will use MySQL server for managing data. For user interface and functionalities, we will use Javascript, PHP, HTML,CSS and some Javascript frameworks such as Nodejs and Express.