

CS 353
DATABASE SYSTEMS
Spring 2018



ÇiftKale
A Football Database System

PROJECT DESIGN REPORT

Prepared By:

Hakan Türkmenoğlu
Orkun Alpar
Selim Fırat Yılmaz
Alper Şahıstan

Table of Contents

1. Revised E/R Model	4
2. Relation Schemas	7
2.1. Person	7
2.2. Agent	8
2.3. Sportsman	9
2.4. Player	10
2.5. Coach	11
2.6. Director	12
2.7. Offer	13
2.8. Bucket	14
2.9. League	15
2.10. Club	16
2.11. WorksFor	17
2.12. LeagueSponsor	18
2.13. ClubSponsor	19
2.14. Match	20
2.15. Stat	21
3. Normalization	22
3.1 League	22
3.2 LeagueCountry	23
4. Functional Components	24
4.1. Use Cases / Scenarios	24
4.2. Algorithms	29
4.3. Data Structures	31
4.4. Use Case Diagram	31
5. User Interface Design and SQL Statements	33
5.1. Login Page	33
5.2. Register Page	34
5.3. Reset Password Page	36
5.4. Leagues List Page	37
5.5. Clubs(Teams) List Page	38
5.6. Players List Page	40
5.7. Transfer Offers List Page	41
5.8. Make Offer For a Player Page	42
5.9. Make Offer To a Team	44
5.10.Coach Information Page	46
5.11.Director Information Page	47
5.12.Change Password Page	48
5.13.Change Username Page	49
5.14.Change Photo Page	50
5.15.Delete Own Account Page	52

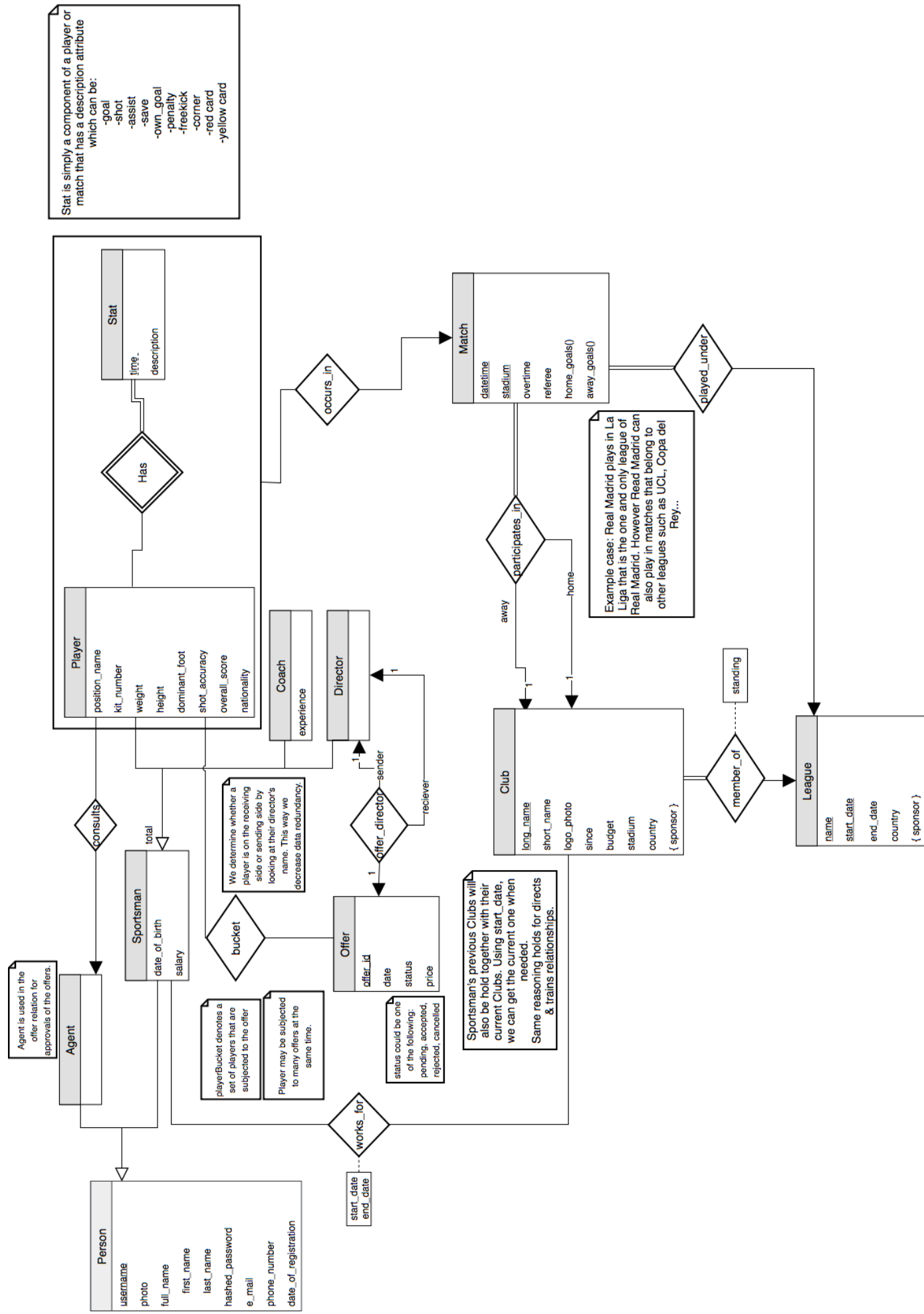
5.16. Your Pending Offers (To accept) Page.....	53
5.17. League Information Page.....	54
5.18. Player Information Page.....	55
5.19. Match History Page.....	57
5.20. Administrator Panel	58
5.21. Agent Information Page.....	59
5.22. Team Information Page.....	60
6. Advanced Database Components	62
6.1. Views.....	62
6.2. Triggers	62
6.3. Constraints	63
7. Implementation Plan	64
8. Website.....	64

1. Revised E/R Model

The E/R model was revised after the feedback from our Tas Arif Usta, Mustafa Can Çavdar and Fuat Basık. The previous E/R model required many changes so we designed from scratch. Changes as follows:

- ✓ Primary keys added for entities.
- ✓ Users are called Person now since user is a reserved word in SQL.
- ✓ Person holds full_name, date_of_registration, photo extra to it's predecessor.
- ✓ Person has two specializations: agent and sportsman.
- ✓ Agent and players are connected via consults relationship
- ✓ Sportsman is an abstraction and the system will not hold a direct instance of it. It contains date_of_birth and salary. It has three specializations: Player, Coach and Director.
- ✓ Sportsman and Club are connected with a works_for relationship(M to M) like in the previous report although it was objected to be wrong. We believe it was a misunderstanding so we added extra explanation notes around. Simply we hold the previous Clubs of the sportsmen.
- ✓ Player holds kit_number, weight, height, dominant foot, shot_accuracy, overall_score extra to it's predecessor. Also all other attributes are removed from it's predecessor.
- ✓ Director's experience_point attribute is removed.
- ✓ Players are connect to a Stats weak entity. Stats hold a time and description(goal, shot, assist, save, own_goal, penalty, freekick, corner, red card, yellow card).
- ✓ An aggregation of a Player and Stats is connected to Match via Occurs_in relationship.
- ✓ Offer system is added. Offer entity and bucket relation. Offer system can be used to offer many combinations of prices and players. An offer is connected to 2 Director via offer_director relationship. Offer is connected to a player via bucket relationship.
- ✓ Clubs now hold logo_photo, budget, stadium, country and sponsors extra to it's predecessor.

- ✓ Leagues now hold start_date, end_date, country and sponsors extra to it's predecessor.
- ✓ Country entity is removed.
- ✓ Club and League has a member_of relation that has standing attribute.
- ✓ Each match is connected to a League via played_under relationship.
- ✓ home_goals and away_goals are now calculated from other relations rather than being held as an attribute.



2. Relation Schemas

2.1. Person

Relational Model

Person(username, photo, first_name, last_name, hashed_password, e_mail, phone_number, date_of_registration)

Functional Dependencies

username → photo, first_name, last_name, hashed_password, e_mail, phone_number, date_of_registration

e_mail → username, photo, first_name, last_name, hashed_password, phone_number, date_of_registration

Candidate Keys

{(username), (e_mail)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Person(  
    username VARCHAR(32) PRIMARY KEY,  
    photo VARCHAR(32),  
    first_name VARCHAR(32) NOT NULL,  
    last_name VARCHAR(32) NOT NULL,  
    hashed_password CHAR(60) NOT NULL,  
    e_mail VARCHAR(32) NOT NULL UNIQUE,  
    phone_number VARCHAR(16) UNIQUE,  
    date_of_registration TIMESTAMP DEFAULT now()  
);
```

2.2. Agent

Relational Model

Agent(agent_username)

Functional Dependencies

None

Candidate Keys

{(agent_username)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Agent(  
    agent_username VARCHAR(32) PRIMARY KEY,  
    FOREIGN KEY (agent_username) REFERENCES Person(username)  
);
```


2.3. Sportsman

Relational Model

Sportsman(sportsman_username, date_of_birth, salary)

Functional Dependencies

sportsman_username → date_of_birth, salary

Candidate Keys

{(sportsman_username)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Sportsman(  
    sportsman_username VARCHAR(32) PRIMARY KEY,  
    date_of_birth DATE NOT NULL,  
    salary INT NOT NULL,  
    FOREIGN KEY (sportsman_username) REFERENCES Person(username)  
);
```

2.4. Player

Relational Model

Player(player_username, position, kit_number, weight, height, dominant_foot, shot_accuracy, overall_score, nationality, agent_username)

Functional Dependencies

player_username → position, kit_number, weight, height, dominant_foot, shot_accuracy, overall_score, agent_username, nationality

Candidate Keys

{(player_username)}

Normal Form

BCNF

Table Definition

```
CREATE TYPE position_t AS ENUM('GK', 'CB', 'LCB', 'RCB', 'LB', 'RB', 'CM', 'LDM', 'RDM', 'CDM', 'CAM', 'LM', 'RM', 'ST', 'CF', 'LW', 'RW');
CREATE TYPE foot_t AS ENUM('left', 'right');
CREATE TABLE Player(
    player_username VARCHAR(32) PRIMARY KEY,
    position position_t,
    kit_number SMALLINT,
    weight SMALLINT,
    height SMALLINT,
    dominant_foot foot_t,
    shot_accuracy REAL CONSTRAINT percentage_real CHECK (shot_accuracy >= 0 AND shot_accuracy <= 1),
    overall_score INT CONSTRAINT percentage_int CHECK (overall_score >= 0 AND overall_score <= 100),
    nationality VARCHAR(32) NOT NULL,
    agent_username VARCHAR(32),
    FOREIGN KEY (player_username) REFERENCES Sportsman(sportsman_username),
    FOREIGN KEY (agent_username) REFERENCES Agent(agent_username)
);
```

2.5. Coach

Relational Model

Coach(coach_username, experience)

Functional Dependencies

coach_username → experience

Candidate Keys

{(coach_username)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Coach(  
    coach_username VARCHAR(32) PRIMARY KEY,  
    experience SMALLINT,  
    FOREIGN KEY (coach_username) REFERENCES Sportsman(sportsman_username)  
);
```

2.6. Director

Relational Model

Director(director_username)

Functional Dependencies

None

Candidate Keys

{(director_username)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Director(  
    director_username VARCHAR(32) PRIMARY KEY,  
    FOREIGN KEY (director_username) REFERENCES  
Sportsman(sportsman_username)  
);
```

2.7. Offer

Relational Model

Offer(offer_id, date, price, status, director_sender, director_receiver)

Functional Dependencies

offer_id → date, price, status, director_sender, director_receiver

Candidate Keys

{(offer_id)}

Normal Form

BCNF

Table Definition

```
CREATE TYPE status_t AS ENUM('pending', 'accepted', 'rejected', 'cancelled');
CREATE TABLE Offer(
    offer_id SERIAL,
    date DATE NOT NULL,
    price INT NOT NULL,
    status status_t NOT NULL,
    director_sender VARCHAR(32) NOT NULL,
    director_receiver VARCHAR(32) NOT NULL,
    PRIMARY KEY (offer_id),
    FOREIGN KEY (director_receiver) REFERENCES Director(director_username),
    FOREIGN KEY (director_sender) REFERENCES Director(director_username),
    CHECK (director_sender <> director_receiver)
);
```

2.8. Bucket

Relational Model

Bucket(offer_id, player_username)

Functional Dependencies

None

Candidate Keys

{(offer_id, player_username)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Bucket(  
    offer_id INT NOT NULL,  
    player_username VARCHAR(32) NOT NULL,  
    PRIMARY KEY (offer_id, player_username),  
    FOREIGN KEY (player_username) REFERENCES Player  
);
```

2.9. League

Relational Model

League(league_name, league_start, league_end, league_country)

Functional Dependencies

(league_name, league_start) → league_end, league_country

(league_name, league_end) → league_start, league_country

league_name → league_country

Candidate Keys

{(league_name, league_start), (league_name, league_end)}

Normal Form

1NF

Table Definition

```
CREATE TABLE League(  
    league_name VARCHAR(32) NOT NULL UNIQUE,  
    league_start DATE NOT NULL,  
    league_end DATE NOT NULL,  
    league_country VARCHAR(32) NOT NULL,  
    UNIQUE (league_name, league_country),  
    UNIQUE (league_name, league_end),  
    PRIMARY KEY (league_name, league_start)  
);
```

2.10. Club

Relational Model

Club(club_name, short_name, logo_photo, since, budget, value, stadium, country, league_name, league_start, standing)

Functional Dependencies

club_name → short_name, logo_photo, since, budget, value, stadium, country, league_name, league_start, standing

Candidate Keys

{(club_name)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Club(  
    club_name VARCHAR(32) PRIMARY KEY,  
    short_name VARCHAR(8) NOT NULL,  
    logo_photo VARCHAR(32) NOT NULL UNIQUE,  
    since DATE NOT NULL,  
    budget INT NOT NULL,  
    value INT NOT NULL,  
    stadium VARCHAR(32) NOT NULL UNIQUE,  
    country VARCHAR(32) NOT NULL,  
    league_name VARCHAR(32) NOT NULL,  
    league_start DATE NOT NULL,  
    standing INT NOT NULL,  
    FOREIGN KEY (league_name, league_start) REFERENCES League  
);
```


2.11. WorksFor

Relational Model

WorksFor(sportsman_username, club_name, start_date, end_date)

Functional Dependencies

sportsman_username, club_name, start_date → end_date

sportsman_username, club_name, end_date → start_date

Candidate Keys

{(sportsman_username, club_name, start_date), (sportsman_username, club_name, end_date)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE WorksFor(  
    sportsman_username VARCHAR(32) NOT NULL,  
    club_name VARCHAR(32) NOT NULL,  
    start_date DATE NOT NULL,  
    end_date DATE,  
    PRIMARY KEY (sportsman_username, club_name, start_date),  
    UNIQUE (sportsman_username, club_name, end_date),  
    FOREIGN KEY (sportsman_username) REFERENCES Sportsman,  
    FOREIGN KEY (club_name) REFERENCES Club  
);
```

2.12. LeagueSponsor

Relational Model

LeagueSponsor(sponsor_name, league_name, league_start)

Functional Dependencies

None

Candidate Keys

{(sponsor_name, league_name, league_start)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE LeagueSponsor(  
    sponsor_name VARCHAR(32) NOT NULL,  
    league_name VARCHAR(32) NOT NULL,  
    league_start DATE NOT NULL,  
    PRIMARY KEY (sponsor_name, league_name, league_start),  
    FOREIGN KEY (league_name, league_start) REFERENCES League  
);
```

2.13. ClubSponsor

Relational Model

ClubSponsor(sponsor_name, club_name)

Functional Dependencies

None

Candidate Keys

{(sponsor_name, club_name)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE ClubSponsor(  
    sponsor_name VARCHAR(32) NOT NULL,  
    club_name VARCHAR(32) NOT NULL,  
    PRIMARY KEY (sponsor_name, club_name),  
    FOREIGN KEY (club_name) REFERENCES Club  
);
```

2.14. Match

Relational Model

Match(datetime, stadium, overtime, referee, home, away, home_goals(), away_goals(), league_name, league_start)

Functional Dependencies

(datetime, stadium) → overtime, referee, home, away, home_goals(), away_goals(), league_name, league_start

(datetime, referee) → stadium, overtime, home, away, home_goals(), away_goals(), league_name, league_start

(datetime, home, away) → stadium, overtime, referee, home_goals(), away_goals(), league_name, league_start

Candidate Keys

{(datetime, stadium), (datetime, referee), (datetime, home, away)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE Match(  
    datetime TIMESTAMP NOT NULL,  
    stadium VARCHAR(32) NOT NULL,  
    overtime SMALLINT NOT NULL,  
    referee VARCHAR(32) NOT NULL,  
    home_goals SMALLINT NOT NULL,  
    away_goals SMALLINT NOT NULL,  
    home VARCHAR(32) NOT NULL,  
    away VARCHAR(32) NOT NULL,  
    league_name VARCHAR(32),  
    league_start DATE,  
    PRIMARY KEY (datetime, stadium),  
    UNIQUE (datetime, referee),  
    UNIQUE (datetime, home, away),  
    FOREIGN KEY (home) REFERENCES Club(club_name),  
    FOREIGN KEY (away) REFERENCES Club(club_name),  
    FOREIGN KEY (league_name, league_start) REFERENCES League,  
    CHECK (home <> away) );
```

2.15. Stat

Relational Model

Stat(player_username, time, description, match_time, stadium)

Functional Dependencies

(player_username, time) → description, match_time, stadium

Candidate Keys

{(player_username, time)}

Normal Form

BCNF

Table Definition

```
CREATE TYPE stat_t AS ENUM('goal', 'shot', 'assist', 'save', 'own_goal', 'penalty', 'freekick',  
'corner', 'red_card', 'yellow_card');
```

```
CREATE TABLE Stat(  
    player_username VARCHAR(32) NOT NULL,  
    time TIMESTAMP NOT NULL,  
    description stat_t NOT NULL,  
    match_time TIMESTAMP NOT NULL,  
    stadium VARCHAR(32) NOT NULL,  
    PRIMARY KEY (player_username, time),  
    FOREIGN KEY (player_username) REFERENCES Player(player_username),  
    FOREIGN KEY (match_time, stadium) REFERENCES Match(datetime, stadium)  
);
```

3. Normalization

All tables are already in BCNF except League table. To normalize League table, we have league_country attribute separated from the League table to reduce data redundancy. Therefore the new definitions of them are as follows:

3.1 League

Relational Model

League(league_name, league_start, league_end)

Functional Dependencies

(league_name, league_start) → league_end

(league_name, league_end) → league_start

Candidate Keys

{(league_name, league_start), (league_name, league_end)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE League(  
    league_name VARCHAR(32) NOT NULL UNIQUE,  
    league_start DATE NOT NULL,  
    league_end DATE NOT NULL,  
    UNIQUE (league_name, league_end),  
    PRIMARY KEY (league_name, league_start)  
);
```

3.2 LeagueCountry

Relational Model

LeagueCountry(league_name, league_country)

Functional Dependencies

league_name \rightarrow league_country

Candidate Keys

{(league_name)}

Normal Form

BCNF

Table Definition

```
CREATE TABLE LeagueCountry(  
    league_name VARCHAR(32) NOT NULL,  
    league_country VARCHAR(32) NOT NULL,  
    PRIMARY KEY (league_name)  
);
```

4. Functional Components

4.1. Use Cases / Scenarios

ÇiftKale is a system that is utilized by different user types, having access to different functionalities. Without including the system administrator role, the superclass of users is namely, *person*¹, which can also be an *agent* or a *sportsman*. Sportsman, which simply is an interface, then gets divided into *player* and *director*. However, *player* and *coach* are not a type of user that interacts with the system. They don't have any use cases.

After all, ÇiftKale has 4 types of users. **Administrator**, **Person**, **Agent** and **Director**, each having access to different functionalities.

[1]: Calling this superclass *user* would be more intuitive and that was our initial choice, however as it turned out to be a reserved word, we changed it to *person* in our E/R diagram and SQL queries. As for the reports, we will use the words *person* and *user* interchangeably.

Administrator

- Supply info/statistics of teams
 - Administrators can supply all the info/statistics of clubs.
- Supply info/statistics of matches
 - Administrators can supply all the info/statistics of matches.
- Supply info/statistics of players
 - Administrators can supply all the info/statistics of players.
- Supply info/statistics of leagues
 - Administrators can supply all the info/statistics of leagues.
- Manipulate statistics
 - Along with adding statistics mentioned above, administrators can also update or delete them.
- View Accounts
 - Administrators can view all accounts that are registered to the system.
- Manipulate Accounts
 - Administrators can edit account information.
- Authorize Accounts
 - Administrators can authorize accounts that are belonging to agents, coaches and directors.
- Delete Accounts
 - Administrators can delete accounts.
- View Leagues
 - Administrators can view leagues, along with information associated with them.

- View Matches
 - Administrators can view specific matches, along with information (teams, result, statistics) associated with them.
- View Match History
 - Administrators can view a page of match results, along with teams, results and dates.
- View Teams
 - Administrators can view teams, along with information (match results, upcoming matches, squad, statistics) associated with them.
- View Player
 - Administrators can view players, their public information and statistics.
- View Coach
 - Administrators can view coaches, their current team, their previous teams, along with public information.
- View Transfer Offers
 - Administrators can view transfer offers, along with information associated with them.
- Share a page via Facebook/Twitter/Google+
 - Administrators can share any page mentioned above via Facebook/Twitter/Google+

Person

- Login
 - Users can login to the system using their user names and passwords. Their passwords will not be stored directly in the database but as hashed.
- Register
 - Person can register to ÇiftKale by supplying a user name, a password, an email address and optionally a mobile phone. Verification will be done using emails or SMSs.
- Reset password
 - User can reset his/her password and select a new one.
- Manage Accounts
 - User can manage his/her account information.
- View Leagues
 - User can view leagues, along with information associated with them.
- View Matches
 - User can view specific matches, along with information (teams, result, statistics, etc) associated with them.
- View Match History
 - User can view a page of match results, along with teams, results and dates.
- View Teams
 - Person can view teams, along with information (match results, upcoming matches, squad, statistics) associated with them.
- View Player
 - Person can view players, their public information and statistics.
- View Coach
 - Person can view coaches, their current team, their previous team, along with public information.
- View Transfer Offers
 - Person can view transfer offers, along with information associated with them.
- Share a page via Facebook/Twitter/Google+
 - User can share any page mentioned above via Facebook/Twitter/Google+

Agent

- Approve a transfer offer
 - Agent can approve a transfer offer directed to and/or involving his/her player (i.e., the player that he/she is the agent of)
- Login
 - Agent can login to the system using their user names and passwords. Their passwords will not be stored directly in the database but as hashed.
- Register
 - Agent can register to ÇiftKale by supplying a user name, a password, an email address and optionally a mobile phone. Verification will be done using emails and SMSs.
- Reset password
 - Agent can reset his/her password and select a new one.
- View Leagues
 - Agent can view leagues, along with information associated with them.
- View Matches
 - Agent can view specific matches, along with information (teams, result, statistics, etc) associated with them.
- View Match History
 - Agent can view a page of match results, along with teams, results and dates.
- View Teams
 - Agent can view teams, along with information (match results, upcoming matches, squad, statistics) associated with them.
- View Player
 - Agent can view players, their public information and statistics.
- View Coach
 - Agent can view coaches, their current club, their previous clubs, along with public information.
- View Transfer Offers
 - Agents can view transfer offers, along with information associated with them..
- Share a page via Facebook/Twitter/Google+
 - Agent can share any page mentioned above via Facebook/Twitter/Google+

Director

- **Make Offer**
 - Director can make an offer to another director. This offer can include 0 to N number of players from each side, along with money from each side.
 - If the offer made exceeds the club budget, this offer will be canceled by the system.
- **Accept Offer**
 - Director can accept the offer proposed to him/her.
 - Either of two sides can cancel the offer or it's canceled by the system due to some reason (e.g. offered player is no longer owned by the team at the time of the offer)
- **Reject Offer**
 - Director can reject the offer proposed to him/her.
- **Login**
 - Director can login to the system using their user name and password. Their passwords will not be stored directly in the database but as hashed.
- **Register**
 - Director can register to ÇiftKale by supplying a user name, a password, an email address and optionally a mobile phone. Verification will be done using emails and SMSs.
- **Reset password**
 - Director can reset his/her password and select a new one.
- **View Leagues**
 - Director can view leagues, along with information associated with them.
- **View Matches**
 - Director can view specific matches, along with information (teams, result, statistics, etc) associated with them.
- **View Match History**
 - Director can view a page of match results, along with teams, results and dates.
- **View Teams**
 - Director can view teams, along with information (match results, upcoming matches, squad, statistics) associated with them.
- **View Player**
 - Director can view players, their public information and statistics.
- **View Coach**
 - Director can view coaches, their current team, their previous team, along with public information.
- **View Transfer Offers**

- Director can view transfer offers, along with information associated with them.
- View Transfers
 - Director can view transfer offers, along with information associated with them.
- Share a page via Facebook/Twitter/Google+
 - Director can share any page mentioned above via Facebook/Twitter/Google+.

4.2.Algorithms

Storing/Showing Photos

For better user experience, we show photos of all persons, coaches, players and also images of clubs, country flags in our user interface. We plan to implement this feature via storing such images in our server's local file system. In our database, we store the image paths of these images. We access these mapped path information via SQL statement and load the image from the local file system of our server. For example, the photo of player stored in “/home/ciftkale/public/img/players/orkun_alpar.png” and we store its path(img/players/orkun_alpar.png) in photo attribute User table.

Table Search/Pagination/Sorting

For better user experience, we combined the search, pagination, sorting functionalities in one table. However, we realized that if we retrieve all data for a table it would be very redundant, reduces performance of both client's browser and database's server. Instead, we only retrieve data for current pagination/search/sorting state. In the beginning and in every state change, we retrieve the data from the Rest API in JSON format according to the pagination/search/sorting state. Namely, it is handled in the back-end without requiring to refreshing the page. It is called **lazy loading**. This is handled in SQL queries in this document below via @searchQuery and @sortQuery variables.

For pagination, we have page number, and # of rows in a page parameters. For sorting we have states of sortable fields, and sorting type(ascending/descending). For searching, we have states of input values for searchable columns of table(of ui). Also, in the beginning, there will be some data retrieved from the backend corresponding to number of pages in the table.

4.3.Data Structures

We use some of SQL's built-in data types such as: SMALLINT, VARCHAR, DATE, TIMESTAMP, SERIAL. The IDs in tables will be auto-incremented with each insert. Also we use the following user-defined types:

status_t: ENUM('pending', 'accepted', 'rejected', 'cancelled')

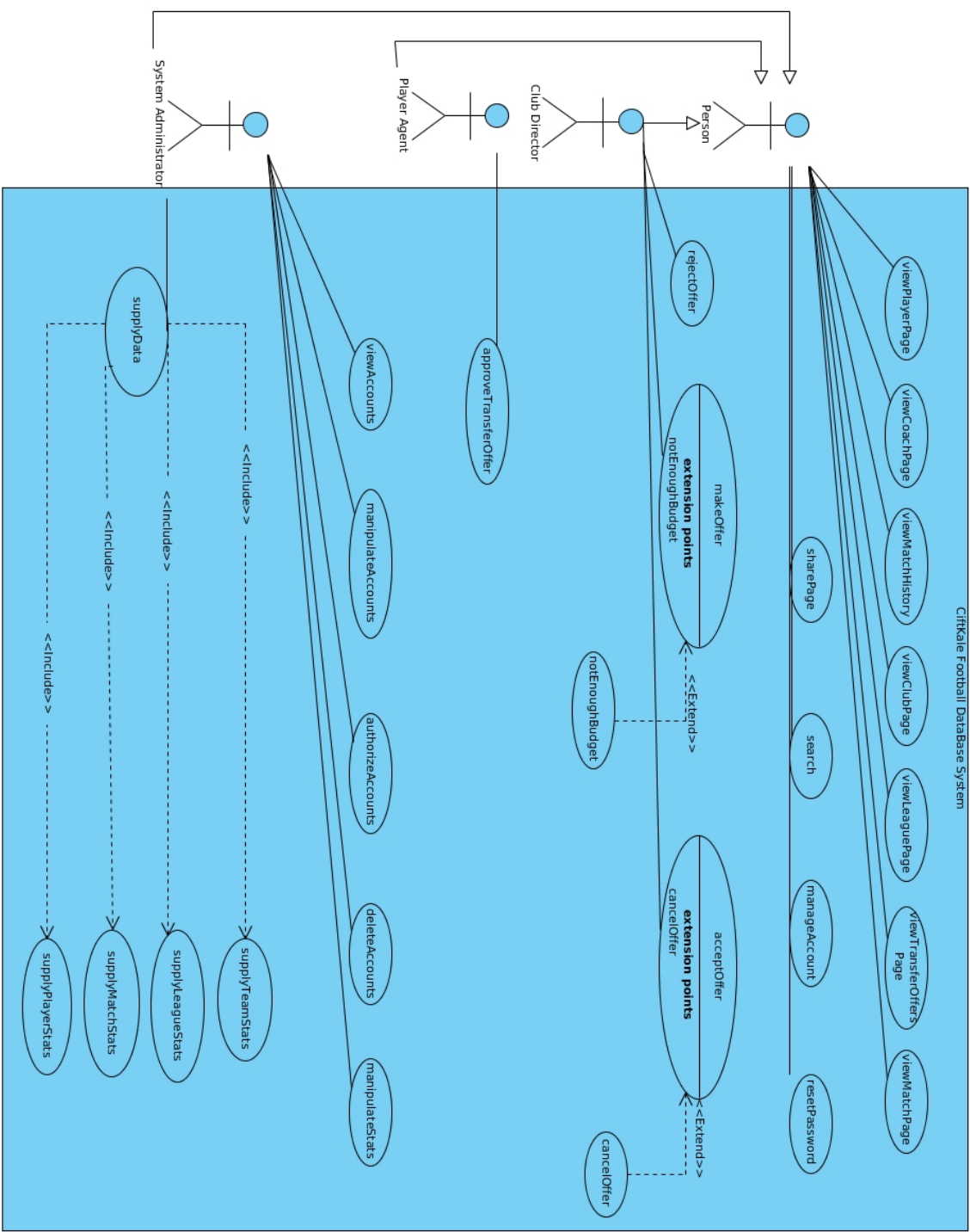
stat_t: ENUM('goal', 'shot', 'assist', 'save', 'own_goal', 'penalty', 'freekick', 'corner', 'red_card', 'yellow_card')

position_t: ENUM('GK', 'CB', 'LCB', 'RCB', 'LB', 'RB', 'CM', 'LDM', 'RDM', 'CDM', 'CAM', 'LM', 'RM', 'ST', 'CF', 'LW', 'RW')

foot_t: ENUM('left', 'right')

4.4.Use Case Diagram

On the following page.



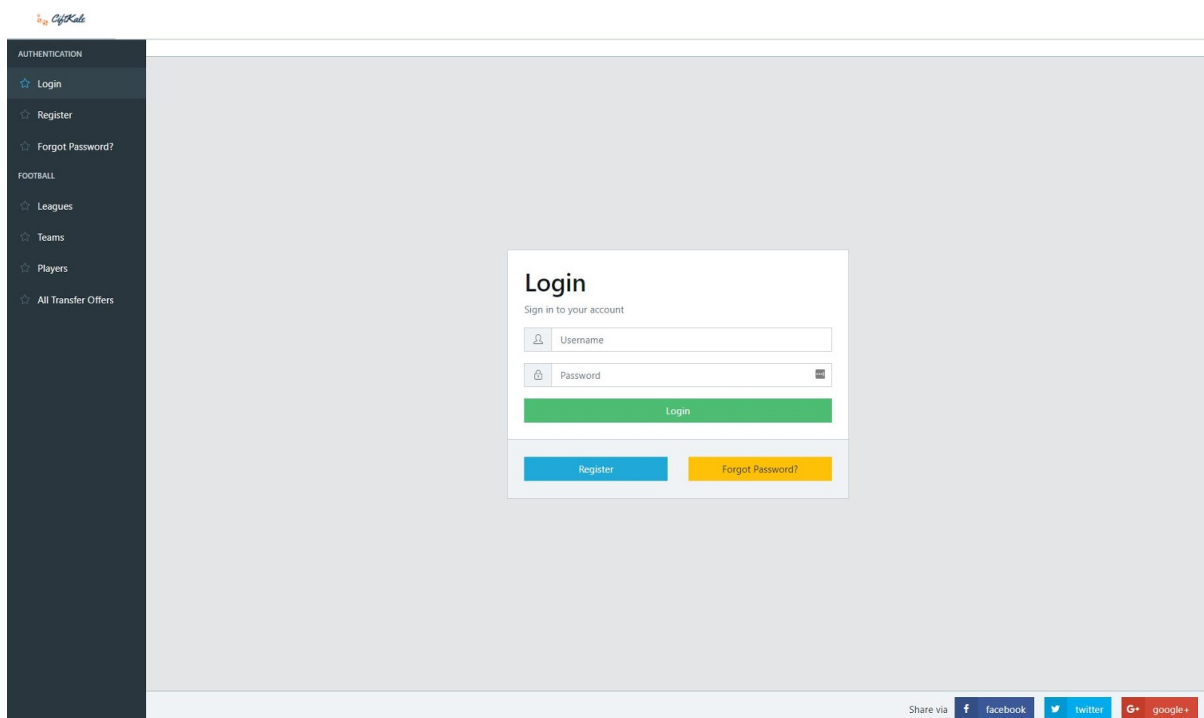
All of the use cases defined requires login to the system. Login requires register. Therefore, these two are not mentioned in the diagram

5. User Interface Design and SQL Statements

We planned to have interactive web pages. Namely we want to load only the required part of the page through Ajax for better user experience and performance. However, we found out that this functionality is hard to realize via wireframes. Thus, we design as many user interface pages in dynamic way using CSS & Javascript functionalities. However, no code for backend functionality is involved yet.

In the view descriptions below, “Accessible by” indicates who can reach this page. “Available actions” indicates what user can do in this page. “Available navigations” indicates where can user go from this page, which are usually the menu and some buttons in the page. “Procedure” indicates how user can do the available actions.

5.1. Login Page



The screenshot shows a web application interface for 'CaféKade'. On the left is a dark sidebar menu with the following items: 'AUTHENTICATION' (containing 'Login', 'Register', and 'Forgot Password?') and 'FOOTBALL' (containing 'Leagues', 'Teams', 'Players', and 'All Transfer Offers'). The main content area has a light gray background and features a white 'Login' form. The form includes the title 'Login', the instruction 'Sign in to your account', and two input fields: 'Username' and 'Password'. Below the inputs is a green 'Login' button. At the bottom of the form are two buttons: a blue 'Register' button and a yellow 'Forgot Password?' button. At the very bottom of the page, there is a footer with the text 'Share via' followed by social media icons for Facebook, Twitter, and Google+.

Accessible by: Anyone entered the website who had not logged in yet.

Available actions: Login

Available navigations: Reset Password, register, leagues list, teams list, players list, transfers list, share via facebook/twitter/google external view

Procedure: User enters his/her username & password which he defined during registration(or last values of such information if they was changed before) to login our system. When user clicks login, if given information is correct, he/she will be redirected to transfers page. If not correct, the following alert will be shown to the user.

Your username or password is wrong!

Inputs: @username, @hashed_password

SQL Statements

```
SELECT * FROM Person WHERE username = @username AND hashed_password = @hashed_password;
```

5.2. Register Page

The screenshot shows the 'Register' page of the CofKale website. On the left is a dark sidebar with a navigation menu. Under the 'AUTHENTICATION' section, there are links for 'Login', 'Register' (which is highlighted), and 'Forgot Password?'. Under the 'FOOTBALL' section, there are links for 'Leagues', 'Teams', 'Players', and 'All Transfer Offers'. The main content area is light gray and contains a white 'Register' form. The form has the title 'Register' and a subtitle 'Create your account as Person'. It includes input fields for 'Full Name', 'Username', 'Email', 'Phone number', 'Password', and 'Repeat password'. Each field has a small icon to its left. The 'Password' and 'Repeat password' fields have a lock icon to their right. Below the form is a green 'Create Account' button. At the bottom of the form are two buttons: a blue 'Login' button and a yellow 'Forgot Password?' button. The footer of the page is light blue and contains the text 'Share via' followed by icons for Facebook, Twitter, and Google+.

Accessible by: Anyone entered the website who had not logged in yet.

Available actions: Create Account

Available navigations: Reset Password, login, leagues list, teams list, players list, transfers list, share via facebook/twitter/google external view

Procedure: User enters his/her desired information(Full name, username, email, password) and re-enters the password again. Then clicks, the “Create account” button. If these information does not satisfy the database constraints or entered passwords are not same, will be shown.(For existing username, “this username is already taken” error is shown.)

Input required

Please provide a valid informations.

Inputs: @username, @photo_path, @first_name, @last_name, @hashed_password, @e_mail, @phone_number

SQL Statements

```
INSERT INTO Person(
username, photo, first_name, last_name, hashed_password, e_mail, phone_number,
date_of_registration)
VALUES (@username, @photo_path, @first_name, @last_name, @hashed_password,
@e_mail, @phone_number, NULL);
```

5.3. Reset Password Page

Forgot Password?
Reset your password

Phone number

Reset Password

Login Register

Share via facebook twitter google+

Accessible by: Anyone entered the website who had not logged in yet.

Available actions: Reset Password

Available navigations: Register, login, leagues list, teams list, players list, transfers list, share via facebook/twitter/google external view

Procedure: For simplicity, user enters his/her phone number. If given phone number does not belong to anybody registered, the following alert will be shown:

There is no user registered with this phone number!

If it matches with a user's phone number, the following SMS message will be sent to him/her:



Thus, he/she can enter with the new password immediately.

Inputs: @phone_number, @new_hashed_password

SQL Statements

UPDATE Person

SET hashed_password = @new_hashed_password

WHERE phone_number = @phone_number;

5.4. Leagues List Page

Country	League	# of Teams	Total Budget
England	Premiere League	32	\$1500000
Spanish	Spanish League	24	\$500000

Accessible by: Anyone entered the website.

Available actions: Change table page, search by country/league name, sort by country, league name, # of teams, total budget

Available navigations: teams list, players list, transfers list, share via facebook/twitter/google external view, leagues in a country, league information page

If not logged in: Reset Password, login, register

If logged in: Change username, change password, change photo (these parts is not shown in the image above but can be found on change username view's section in menu)

If Director: Your Pending Offers Page

Procedure: User can see the all leagues and their information(total budget, # of teams, country, country flag) via the table. Details are described in Algorithms->Table Search/ Pagination/Sorting section. If user clicks any country in country column, the table will be filtered by this country. If user clicks a league in league name column, the user will be redirected to this league's information page.

Inputs: @searchQuery, @orderQuery, @paginationQuery

SQL Statements

```
SELECT * FROM League NATURAL JOIN LeagueCountry WHERE (@searchQuery) ORDER BY
@orderQuery LIMIT @paginationQuery;
```

5.5. Clubs(Teams) List Page

[illegible]

Accessible by: Anyone entered the website.

Available actions: Change table page, search by country/league name, team name, team coach, team director, sort by league standing, country, league name, team name, team coach, team director, and team budget.

Available navigations: leagues list, players list, transfers list, share via facebook/
twitter/google external view, league information page, team information page, coach
information page, director information page

If not logged in: Reset Password, login, register

If logged in: Change username, change password, change photo

If Director: Your Pending Offers Page

Procedure: User can see the all clubs(or teams) via the table. Details are described in Algorithms->Table Search/Pagination/Sorting section. User can filter the table by country by clicking a country in country column. User can access league information page by clicking a league in league column. User can access team information page by clicking a team in team name column. User can access coach information page by clicking a coach in coach column. User can access director information page by clicking a director in director column.

Note: Naming difference (Teams, clubs) creates an inconsistency but will be fixed by changing all “teams” to “clubs” during implementation.

Inputs: @searchQuery, @orderQuery, @paginationQuery

SQL Statements

```
SELECT cl.*, c.*, d.* FROM Club NATURAL JOIN League NATURAL JOIN LeagueCountry cl,  
Director d, Coach c, WorksFor wf1, WorksFor wf2  
WHERE d.director_username = wf1.sportsman_username AND  
      c.coach_username = wf2.sportsman_username AND  
      wf1.end_date IS NULL AND  
      wf2.end_date IS NULL AND  
      cl.club_name = wf1.club_name AND  
      cl.club_name = wf2.club_name AND  
      (@searchQuery)  
ORDER BY @orderQuery LIMIT @paginationQuery;
```

5.6. Players List Page

Team	Nation	Player	Overall
Name		Name	
Barcelona	Spanish	Sergi Roberto	94

Previous Page 1 of 1 10 rows Next

Share via facebook twitter google+

Accessible by: Anyone entered the website.

Available actions: Change table page, search by team name, player's nation, player name, sort by country, league name, # of teams, total budget, team information page, player information page.

Available navigations: leagues list, teams list, transfers list, share via facebook/twitter/google external view

If not logged in: Reset Password, login, register

If logged in: Change username, change password, change photo

If Director: Your Pending Offers Page

Procedure: User can see/search/sort/navigate all(or filtered) players via the table. Functionality details are described in Algorithms->Table Search/Pagination/Sorting section. User can access team information page by clicking a team in team name column. User can filter the table by clicking a country in country column. User can access player information page by clicking a player in player name column.

Inputs: @searchQuery, @sortQuery, @paginationQuery


```
SELECT * FROM Player p, Club NATURAL JOIN CurrentOccupations co
WHERE p.player_username = co.sportsman_username AND
      (@searchQuery)
ORDER BY @sortQuery LIMIT @paginationQuery;
```

	From Club		To Club		Offer Info			
	Name	Director	Name	Agent	Player(s)	Offer		Date
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	All ▾	
Arsenal	Alper Şahistan		Barcelona	Hakan Türkmenoğlu	Sabri Sançlı, Hasan Şaş	\$19M + Volkan Demirel	Pending	2018-01-20
Arsenal	Alper Şahistan		Barcelona	Hakan Türkmenoğlu	Mesut Özil	\$43M	Accepted	2018-01-13
Previous		Page 1 of 1		10 rows ▾		Next		

Available actions: Change table page, search by offerer club's name, offerer director's name, target club's name, target player(s)'s agent, target players, offer status, sort by offerer club's name, offerer director's name, target club's name, target player(s)'s agent, target players, offer status, target players & money amount, and date of offer.

If Director: Your Pending Offers Page

Procedure: User can see/search/sort/navigate all(or filtered) offers via the table. Functionality details are described in Algorithms->Table Search/Pagination/Sorting

section. User can filter the table by clicking offerer club's name, offered players. User can access director's page by clicking a director in any director column. User can access agent information page by clicking agent in agent column. User can access player information page by clicking a player in player(s) column.

SQL Statements

```
WITH AllPlayers AS (SELECT * FROM Offer o LEFT JOIN Bucket b ON b.offer_id =  
o.offer_id LEFT JOIN PlayerDirector pd ON pd.player_username = b.player_username)  
SELECT DISTINCT ap.* FROM AllPlayers ap  
WHERE ap.director_username = ap.director_sender AND  
(@searchQuery) ORDER BY @sortQuery LIMIT @paginationQuery;
```

```
WITH AllPlayers AS (SELECT * FROM Offer o LEFT JOIN Bucket b ON b.offer_id =  
o.offer_id LEFT JOIN PlayerDirector pd ON pd.player_username = b.player_username)  
SELECT DISTINCT ap.* FROM AllPlayers ap  
WHERE ap.director_username = ap.director_receiver AND  
(@searchQuery) ORDER BY @sortQuery LIMIT @paginationQuery;
```

5.8. Make Offer For a Player Page

Make Offer: Sergi Roberto

You can make offer via money and also optionally by adding some players from your team to the bucket.

Select Players in your team to send as bucket

Player #1
Player #2
Player #3
Player #4

Share via [facebook](#) [twitter](#) [google+](#)

Accessible by: Directors

Available actions: Make offer for a player

Available navigations: leagues list, teams list, players list, share via facebook/twitter/google external view, change username, change password, change photo, your pending offers

Procedure: The director enters the money amount and selects players in his/her team to be added to the transfer bucket. If entered money amount is larger than his/her team's budget, the following alert will be shown.

Your team has not enough budget

Otherwise, he will be redirected to the transfers list page.

SQL Statements

-- Get the player's director_name, store it as @target_director_username

```
SELECT director_username FROM PlayerDirector WHERE player_username =  
@target_player_username;
```

-- Get the max offer_id and increment it by one, store it as @new_offer_id

```
SELECT MAX(offer_id) FROM Offer;
```

-- Create a new offer record

```
INSERT INTO Offer(offer_id, date, price, status, director_sender, director_receiver)  
VALUES(@new_offer_id, NULL, @price, 'pending', @my_username,  
@target_director_username);
```

-- These are executed in a for loop

```
INSERT INTO Bucket(offer_id, player_username) VALUES(@new_offer_id,  
@my_player_username);
```

-- End for loop

5.9. Make Offer To a Team

The screenshot shows the 'Make Offer: Barcelona' form within the CapKa application. On the left is a dark sidebar with navigation links under 'ACCOUNT' (Change Password, Delete Own Account, Change Photo, Change Username) and 'FOOTBALL' (Leagues, Teams, Players, All Transfer Offers, Your Pending Offers). The main content area features a form titled 'Make Offer: Barcelona' with the instruction: 'You can make offer via money and also optionally by adding some players from your team to the bucket.' The form includes a currency selector set to '\$' and a 'Money Amount' input field. Below this are two selection sections: 'Select Players in target team to get as bucket' with a list of 'Target Team Player #1' through '#4', and 'Select Players in your team to send as bucket' with a list of 'Player #1' through '#4'. A green 'Make offer' button is at the bottom of the form. At the bottom right of the page, there is a 'Share via' section with icons for Facebook, Twitter, and Google+.

Accessible by: Directors

Available actions: Make offer to a team

Available navigations: leagues list, teams list, players list, share via facebook/twitter/google external view, change username, change password, change photo, your pending offers Page

Procedure: The director enters the money amount and selects players in his/her team to be added to the transfer bucket and players in target team he/she wants to transfer in return. If entered money amount is larger than his/her team's budget, the following alert will be shown.

Your team has not enough budget

Otherwise, he will be redirected to the transfers list page.

SQL Statements

```

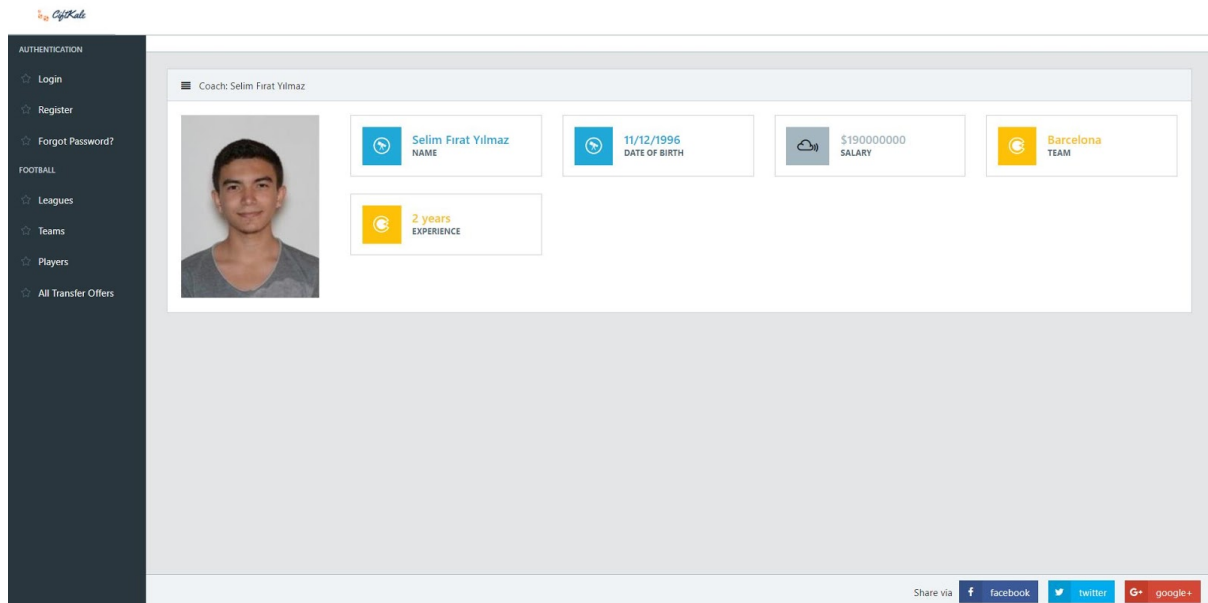
-- Get the target club's director username and store it as @target_director_username
SELECT d.director_username FROM Club NATURAL JOIN CurrentOccupations co, Director
d WHERE d.director_username = co.sportsman_username AND co.club_name =
@target_club_name;

-- Get the max offer_id and increment it by one, store it as @new_offer_id
SELECT MAX(offer_id) FROM Offer;

-- Create a new offer record
INSERT INTO Offer(offer_id, date, price, status, director_sender, director_receiver)
VALUES(@new_offer_id, NULL, @price, 'pending', @my_username,
@target_director_username);
-- These are executed in a for loop
INSERT INTO Bucket(offer_id, player_username) VALUES(@new_offer_id,
@my_player_username);
-- End for loop
-- These are executed in a for loop
INSERT INTO Bucket(offer_id, player_username) VALUES(@new_offer_id,
@target_player_username);
-- End for loop

```

5.10.Coach Information Page



Accessible by: Anyone entered the website.

Available actions: -

Available navigations: leagues list, teams list, players list, share via facebook/twitter/google external view, coach's team

If not logged in: Reset Password, login, register

If logged in: Change username, change password, change photo

If Director: Your Pending Offers Page

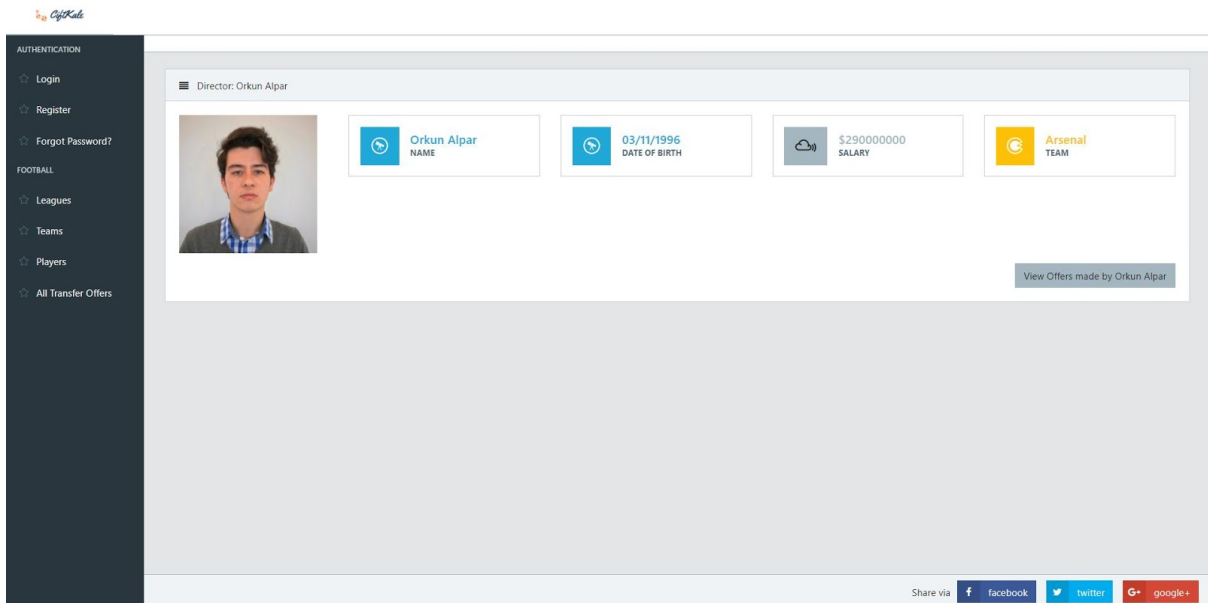
Procedure: User can see the information such as coach's name, date of birth, salary, team, and experience and can navigate to coach's team's page via the team link.

Inputs: @username

SQL Statements

```
SELECT * FROM Coach WHERE coach_username = @username;
```

5.11.Director Information Page



Accessible by: Anyone

Available actions: -

Available navigations: leagues list, teams list, players list, share via facebook/twitter/google external view, director's team, transfer offers made by director

If not logged in: Reset Password, login, register

If logged in: Change username, change password, change photo

If Director: Your Pending Offers Page

Procedure: User can navigate to director's team's page via the team link. Also, user can navigate to transfer offers made by director via "View Offers made by #directorname" button. This navigation will redirect to Transfer Offers List Page with director filtering as below:

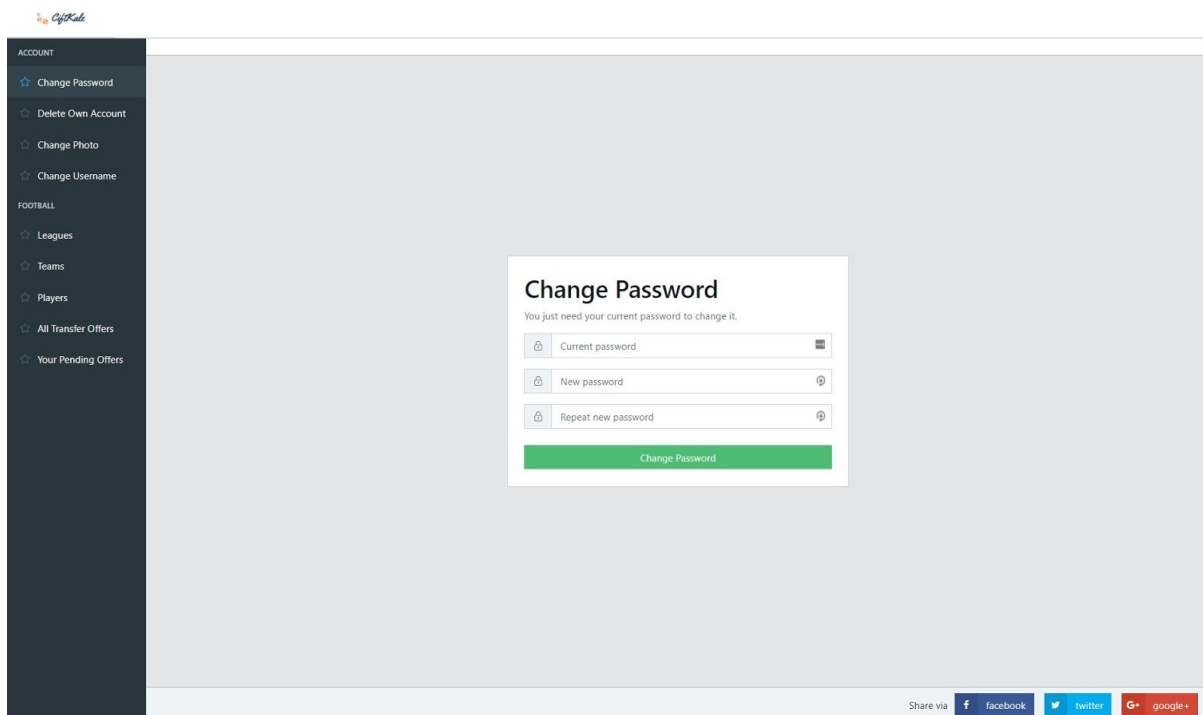
Transfer Offers					
From Club			To Club		
	Name	Director		Name	
	<input type="text"/>	Alper Şahistan		<input type="text"/>	
	Arsenal	Alper Şahistan		Barcelona	Hakan T
	Arsenal	Alper Şahistan		Barcelona	Hakan T

Inputs: @username

SQL Statements

```
SELECT * FROM Director WHERE director_username = @username;
```

5.12.Change Password Page



The screenshot shows the 'Change Password' page of the CofKale website. On the left is a dark sidebar with a navigation menu. Under the 'ACCOUNT' section, 'Change Password' is highlighted. Other options include 'Delete Own Account', 'Change Photo', and 'Change Username'. Under the 'FOOTBALL' section, there are links for 'Leagues', 'Teams', 'Players', 'All Transfer Offers', and 'Your Pending Offers'. The main content area has a light gray background. In the center is a white box titled 'Change Password' with the instruction 'You just need your current password to change it.' Below this are three input fields: 'Current password', 'New password', and 'Repeat new password', each with a lock icon on the right. At the bottom of the box is a green button labeled 'Change Password'. At the bottom right of the page, there is a 'Share via' section with icons for Facebook, Twitter, and Google+.

Accessible by: Anyone who logged in.

Available actions: Change Password

Available navigations: Delete own account, change photo, change username, leagues list, teams list, players list, transfers list, share via facebook/twitter/google external view

Procedure: User enters current password which he defined during registration(or last values of such information if they was changed before) and new password he would like to use. When user clicks Change Password button, if given information is correct, he/she will be shown the following alert:

Your password is succesfully changed!

If not correct, the following alert will be shown to the user:

Your current password is not correct!

Inputs: @old_hashed_password, @new_hashed_password

Sql Statements

UPDATE Person

SET hashed_password = @new_hashed_password

WHERE hashed_password @old_hashed_password;

5.13.Change Username Page

The screenshot shows the 'Change Username' page in the CityKade application. On the left is a dark sidebar with a navigation menu. Under the 'ACCOUNT' section, the options are 'Change Password', 'Delete Own Account', 'Change Photo', and 'Change Username' (which is highlighted). Under the 'FOOTBALL' section, the options are 'Leagues', 'Teams', 'Players', 'All Transfer Offers', and 'Your Pending Offers'. The main content area is light gray and contains a white form titled 'Change Username'. The form has the subtitle 'Change your username' and two input fields: 'New Username' and 'Current Password'. Below the fields is a green button labeled 'Change your username'. At the bottom right of the page, there is a 'Share via' section with icons for Facebook, Twitter, and Google+.

Accessible by: Anyone who logged in.

Available actions: Change Username

Available navigations: Delete own account, change photo, change username, leagues list, teams list, players list, transfers list, share via facebook/twitter/google external view

If Director: Your Pending Offers Page

Procedure: User enters current password which he defined during registration(or last values of such information if they was changed before) and new username he would like to use. When user clicks Change Username button, if given information is correct, he/she will be shown the following alert:

Your username is succesfully changed!

If not correct, the following alert will be shown to the user:

Your current password is not correct!

Inputs: @new_username, @old_username, @hashed_password

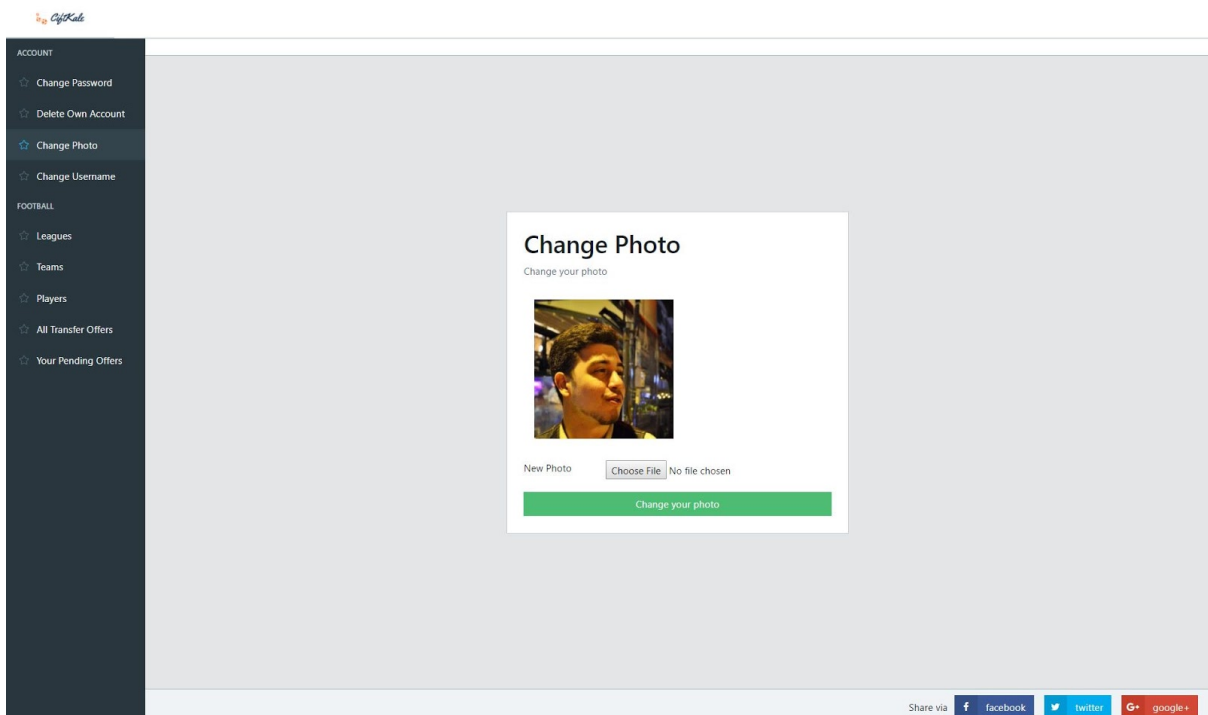
Sql Statements

UPDATE Person

SET username = @new_username

WHERE username = @old_username AND hashed_password = @hashed_password;

5.14.Change Photo Page



Accessible by: Anyone who logged in.

Available actions: Change Photo

Available navigations: Delete own account, change password, change username, leagues list, teams list, players list, transfers list, share via facebook/twitter/google external view

If Director: Your Pending Offers Page

Procedure: User select his new photo via file selector. When user clicks Change your photo button, if selected photo is valid, following alert will be shown to the user:

Your photo is succesfully changed!

If not valid, the following alert will be shown to the user:

The photo you are trying to upload is not valid!

Inputs: @new_photo_path, @username

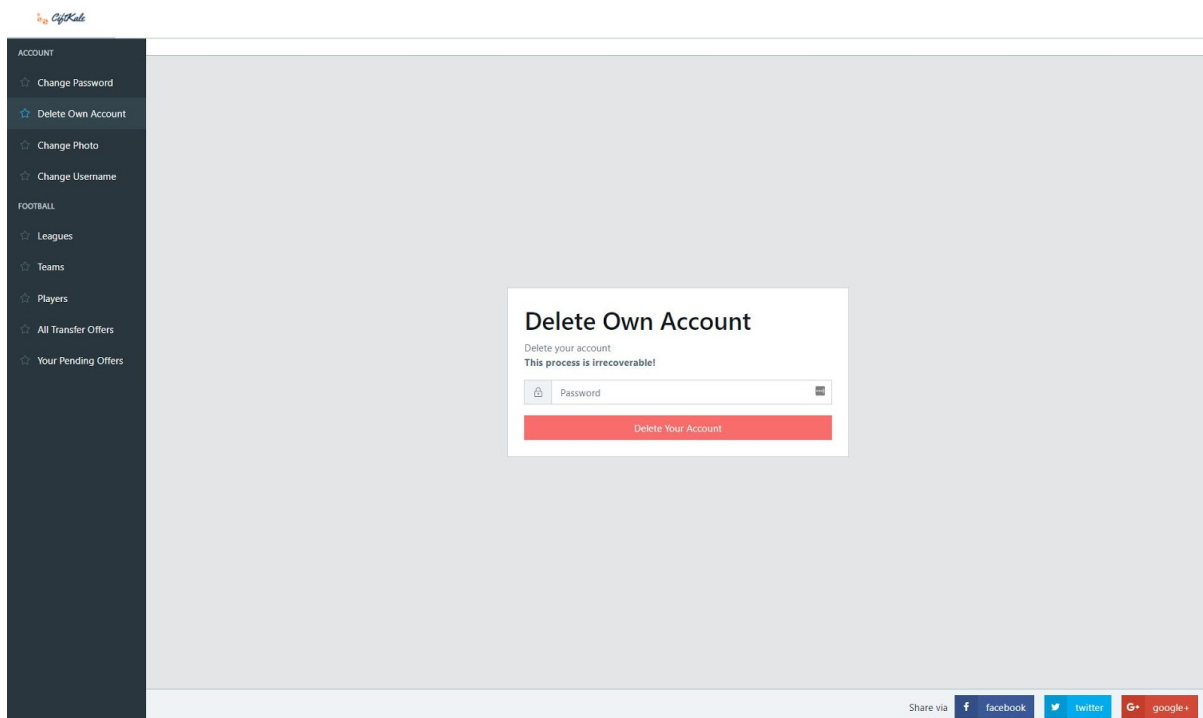
Sql Statements

UPDATE Person

SET photo = @new_photo_path

WHERE username = @username;

5.15.Delete Own Account Page



Accessible by: Anyone who logged in.

Available actions: Delete own account

Available navigations: Change password, change photo, change username, leagues list, teams list, players list, transfers list, share via facebook/twitter/google external view
If Director: Your Pending Offers Page

Procedure: User enters current password which he defined during registration(or last values of such information if they was changed before) and new username he would like to use. When user clicks Change Username button, if given information is correct, he/ she will be redirected to transfers list page as logged out. If not correct, the following alert will be shown to the user:

Your current password is not correct!

Inputs: @username, @hashed_password

Sql Statements

-- check for current password query

SELECT FROM Person

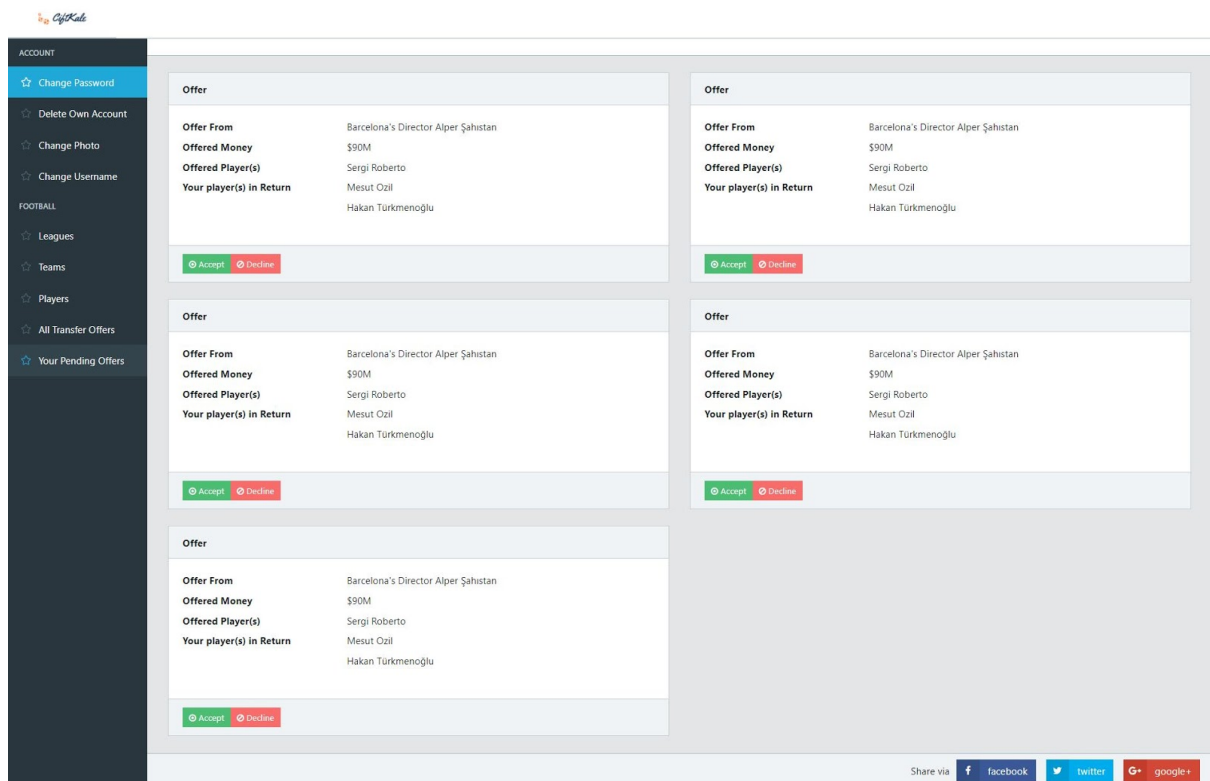
WHERE username = @username AND hashed_password = @hashed_password;

-- delete query

DELETE FROM Person

WHERE username = @username AND hashed_password = @hashed_password;

5.16. Your Pending Offers (To accept) Page



Accessible by: Directors and Agents

Available actions: Accept/Decline an offer

Available navigations: Delete own account, change photo, change username, leagues list, teams list, players list, transfers list, share via facebook/twitter/google external view, your pending offers

Procedure: Director/Agent can see the information such as the offerer, offered money, offered players and the players offerer wants in return. The director/agent can accept/decline the offer via the buttons underneath the offer.

Inputs: @username

Sql Statements

```
WITH AllPlayers AS (SELECT * FROM Offer o LEFT JOIN Bucket b ON b.offer_id =  
o.offer_id LEFT JOIN PlayerDirector pd ON pd.player_username = b.player_username)  
SELECT DISTINCT ap.* FROM AllPlayers ap  
WHERE ap.director_username = ap.director_receiver AND  
ap.director_username = @username;
```

5.17. League Information Page

The screenshot displays the 'Premiere League' information page. The left sidebar contains navigation links for Authentication (Login, Register, Forgot Password?), Football (Leagues, Teams, Players, All Transfer Offers), and a footer with social media links. The main content area is titled 'Premiere League' and includes three cards: 'Premiere League LEAGUE NAME', '2017 Spring CURRENT SEASON', and 'England COUNTRY'. Below these is a 'Clubs List' table with the following data:

League			Team				
Standing	Country	Name		Name	Coach	Director	Budget
1	England	Premiere League	ARS	Arsenal	Hakan Türkmenoğlu	Alper Şahistan	\$1500000

Below the table is a 'Sponsors' section featuring logos for New Balance, Pepsi, and Toyota. At the bottom right, there are social media share buttons for Facebook, Twitter, and Google+.

Accessible by: Anyone entered the website.

Available actions: -

Available navigations: leagues list, teams list, players list, share via facebook/twitter/google external view, director's team, transfer offers made by director

If not logged in: Reset Password, login, register

If logged in: Change username, change password, change photo

If Director: Your Pending Offers Page

Procedure: User can see the sponsors of the league, information of the league such as its clubs, league name, current season, and its country

Inputs: @searchQuery, @sortQuery, @paginationQuery

Sql Statements

-- Get Club list

```
SELECT * FROM Club NATURAL JOIN League NATURAL JOIN LeagueCountry
WHERE (@searchQuery) ORDER BY @sortQuery LIMIT @paginationQuery;
```

-- Get sponsor list

```
SELECT * FROM LeagueSponsor WHERE (@searchQuery);
```

5.18.Player Information Page

The screenshot shows a web application interface for a player's information. On the left is a dark sidebar with navigation links under 'ACCOUNT' (Change Password, Delete Own Account, Change Photo, Change Username) and 'FOOTBALL' (Leagues, Teams, Players, All Transfer Offers, Your Pending Offers). The main content area is titled 'Player: Sergi Roberto'. It features a player profile card with a photo, name, and salary (\$15,000,000). To the right are cards for 'Defence' (POSITION NAME), '20' (KIT NUMBER), and 'Right' (DOMINANT FOOT). Below these are cards for 'Barcelona' (TEAM) and 'Hakan Türkmenoğlu' (AGENT). At the bottom right of the main area are 'View Offers' and 'Make Offer' buttons. A 'Statistics' table is located below the main content area.

Statistics	
Total goals	24
Total shoots	757
Total assists	43
Total yellow cards	123
Total red cards	24

At the bottom of the page, there is a 'Share via' section with icons for Facebook, Twitter, and Google+.

Accessible by: Anyone entered the website.

Available actions: -

Available navigations: leagues list, teams list, players list, share via facebook/twitter/google+ external view, agent's players' information pages, View offers

If not logged in: Reset Password, login, register

If logged in: Change username, change password, change photo

If Director: Your Pending Offers Page, Make offer(its button only shown to director)

Procedure: User can see the player's information such as name, preferred position, kit number, dominant foot, salary, team, agent and its statistics(total goals/shoots/assists/yellow cards/red cards(these statistics will be revised)) User can access player's agent's information page via clicking its agent. User can access player's team's information page via clicking its team. A director can make an offer to this player by clicking "Make offer" button and redirected to "Make offer for a player" page. User can access offers related to this player by clicking "View offers" button and redirected to transfer offers page in which table is filtered by the player's name just as is in the director's information page.

Inputs: @username

SQL Statement

-- Player information

```
SELECT * FROM Player WHERE player_username = @username;
```

-- Player stats (each of rows indicates occurrence of 1 entity(e.g. goal))

```
SELECT count(*) FROM Stat WHERE player_username = @username AND description = 'goal';
```

```
SELECT count(*) FROM Stat WHERE player_username = @username AND description = 'assist';
```

```
SELECT count(*) FROM Stat WHERE player_username = @username AND description = 'shot';
```

```
SELECT count(*) FROM Stat WHERE player_username = @username AND description = 'yellow_card';
```

```
SELECT count(*) FROM Stat WHERE player_username = @username AND description = 'red_card';
```


5.19.Match History Page

Match History						
▼ Home Team	▼ Score	▼ Away Team	▼ Stadium	▼ Date	▼ Time	▼ Referee
<input type="text" value="Q Search..."/>	<input type="text" value="Q Search..."/>	<input type="text" value="Q Search..."/>	<input type="text" value="Q Search..."/>	<input type="text" value="Q Search..."/>	<input type="text" value="Q Search..."/>	<input type="text" value="Q Search..."/>
Fenerbahce	6-0	Galatasaray	Sukru Saracoglu	06.11.2002	20:00	Mustafa Culcu
Liverpool	8-0	Besiktas	Anfield	06.11.2007	21:45	Markus Merk
Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7
Cell 8	Cell 9	Cell 10	Cell 11	Cell 12	Cell 13	Cell 14

Accessible by: Anyone entered the website.

Available actions: -

Available navigations: leagues list, teams list, players list, share via facebook/twitter/google+ external view, agent's players' information pages, View offers

If not logged in: Reset Password, login, register

If logged in: Change username, change password, change photo

If Director: Your Pending Offers Page, Make offer(its button only shown to director)

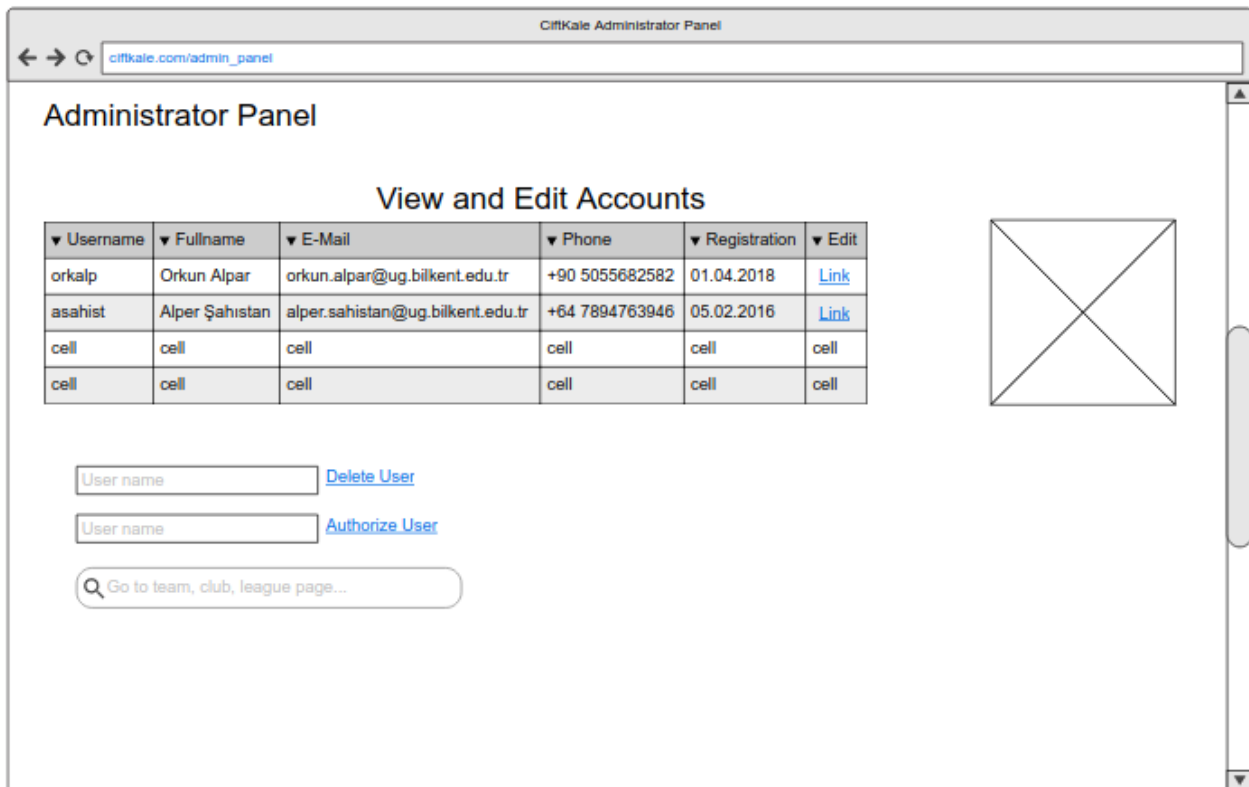
Procedure: User can see home team, away team, score, stadium, date, time and referee. Each of these information may be filtered by the user and paginated.

Inputs: @searchQuery, @paginationQuery, @sortQuery

SQL Statement

```
SELECT * FROM Match WHERE (@searchQuery) ORDER BY @sortQuery LIMIT @sortQuery;
```

5.20.Administrator Panel



The screenshot shows a web browser window titled "CiftKale Administrator Panel" with the URL "ciftkale.com/admin_panel". The page has a header "Administrator Panel" and a main section titled "View and Edit Accounts". It contains a table with user information and a search bar.

▼ Username	▼ Fullname	▼ E-Mail	▼ Phone	▼ Registration	▼ Edit
orkalp	Orkun Alpar	orkun.alpar@ug.bilkent.edu.tr	+90 5055682582	01.04.2018	Link
asahist	Alper Şahistan	alper.sahistan@ug.bilkent.edu.tr	+64 7894763946	05.02.2016	Link
cell	cell	cell	cell	cell	cell
cell	cell	cell	cell	cell	cell

Below the table, there are two input fields for "User name" with buttons "Delete User" and "Authorize User". At the bottom, there is a search bar with the placeholder text "Go to team, club, league page..." and a magnifying glass icon.

Accessible by: Only by the administrator of the website.

Available actions: -

Available navigations: None

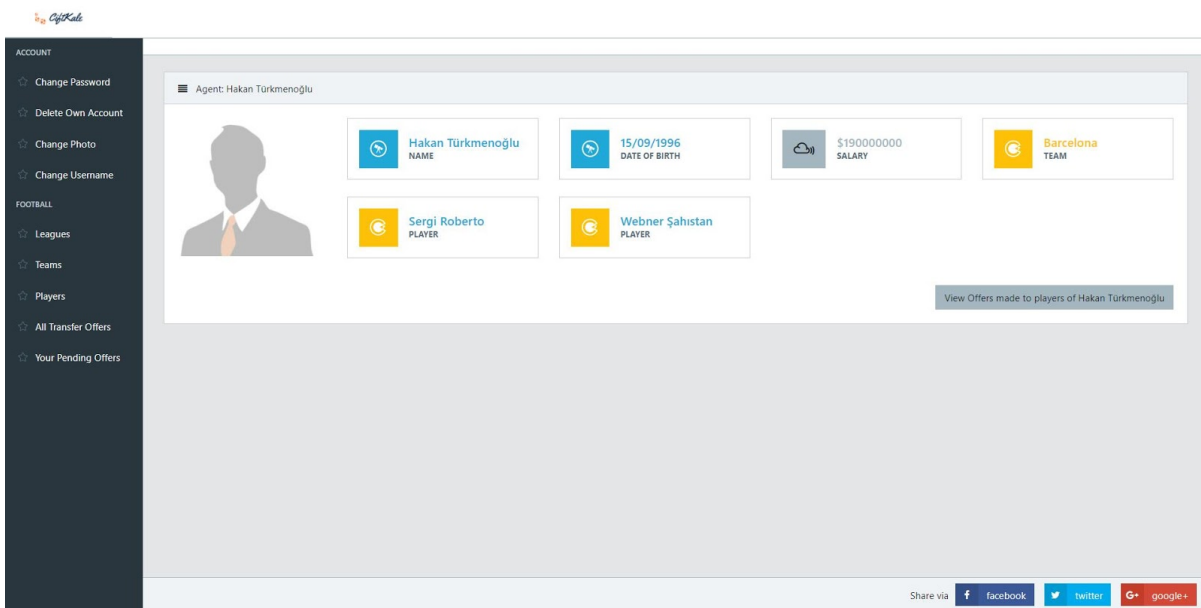
Procedure: Admin can see account information, edit and delete users. Additionally the admin may give special permissions to the users. Also it's possible to filter account information.

Inputs: @searchQuery, @paginationQuery, @sortQuery, @editQuery1, @editQuery2

SQL Statement

```
SELECT * FROM Person WHERE (@searchQuery) ORDER BY @sortQuery LIMIT @sortQuery;  
UPDATE Person  
SET (@editQuery1)  
WHERE (@editQuery2);
```

5.21.Agent Information Page



Accessible by: Anyone entered the website.

Available actions: -

Available navigations: leagues list, teams list, players list, share via facebook/twitter/google+ external view, agent's players' information pages

If not logged in: Reset Password, login, register

If logged in: Change username, change password, change photo

If Director: Your Pending Offers Page

Procedure: User can see the agent's information such as name, date of birth, salary, and its players. User can access the offered made to agent's players of interest by the "View offers made to players of #agentname" button. User can access to agent's players' information pages via clicking players' names.

Inputs: @username

Sql Statement

```
SELECT * FROM Agent WHERE agent_username = @username;
```

5.22.Team Information Page

The screenshot displays the 'Team Information Page' for FC Barcelona. The page is divided into three main sections: Team Details, Players, and Sponsors.

Team Details: This section includes the FC Barcelona crest and several key pieces of information:

- Team Name:** Barcelona(BAR)
- League Name:** Liga BBVA
- Date of Foundation:** 11/12/1950
- Stadium:** Camp Nou
- Transfer Budget:** \$195000
- Coach:** Ernesto Valverde
- Director:** Josep Maria Bartomeu i Floreta

A 'Make Offer' button is located at the bottom right of this section.

Players: This section contains a table listing the team's players. The table has columns for Team, Name, Nation, Player, and Overall.

Team	Name	Nation	Player	Overall
Barcelona				
Barcelona		Spanish	Sergi Roberto	94

At the bottom of the table, there is a pagination bar showing 'Page 1 of 1' and '10 rows'.

Sponsors: This section displays the logos of the team's sponsors: New Balance, Pepsi, and Toyota.

At the bottom of the page, there is a 'Share via' section with links to Facebook, Twitter, and Google+.

Accessible by: Anyone entered the website.

Available actions: -

Available navigations: leagues list, teams list, players list, share via facebook/twitter/google external view, director's team, team's director's page, team's coach's page, make offer to team page

If not logged in: Reset Password, login, register

If logged in: Change username, change password, change photo

If Director: Your Pending Offers Page (shown twice in the screenshot above due to full snapshot feature of browser's extension)

Procedure: User can see the sponsors of the team, information of the team such as its players and their information(name, nation, overall score, name, photo), league name, date of foundation, stadium, transfer budget, coach, director.

Inputs: @club_name

Sql Statements

-- Get club information

```
SELECT * FROM Club where club_name = @club_name;
```

-- Get player list

```
SELECT * FROM Player p, CurrentOccupations co
```

```
WHERE p.player_username = co.sportsman_username AND  
      co.club_name = @club_name;
```

-- Get sponsor list

```
SELECT * FROM ClubSponsor WHERE club_name = @club_name;
```

6. Advanced Database Components

6.1. Views

A lot of the times we need to access a Sportsman's current club name. This is addressed by the CurrentOccupations view:

```
CREATE VIEW CurrentOccupations AS (  
  SELECT * FROM WorksFor  
  WHERE end_date IS NULL  
);
```

We have also declared PlayerDirector view that matches a player's current director.

```
CREATE VIEW PlayerDirector AS (  
  SELECT p.player_username, d.director_username  
  FROM player p, director d, CurrentOccupations cop, CurrentOccupations cod  
  WHERE p.player_username = cop.sportsman_username AND  
         d.director_username = cod.sportsman_username AND  
         cop.club_name = cod.club_name  
);
```

6.2. Triggers

- When a match is inserted, deleted, updated related Club, Player and League will be updated.
- When an offer is accepted if there are another offers involving Players in the accepted offer relation those offers will be cancelled.
- When a Player, Coach or Director is dropped corresponding Sportsman and Person Tables will be dropped as well. And the corresponding values in works_for tables will be null.
- When a Player is deleted related stats will be deleted as well as that player's corresponding consults relation will be deleted.
- When a Player is transferred to another team his salary from sportsman table position_name and kit_number from player table will be updated.
- If Budget of a club is decreased their offers that they offer money will be checked if they are lower than the budget system will drop the corresponding offer.
- When a Sportsman is inserted his works_for relation will be updated according to his club.

6.3. Constraints

- A budget of a team cannot be exceeded by an offer made by director of that team.
- Clubs and Leagues cannot be deleted.
- There can be at most 10000 leagues in the system.
- There can be at most 50000 clubs in the system.
- There can be at most 1000000 players in the system.
- There can be at most 10000000 users in the system.
- Only directors can initiate, view, accept, decline or cancel offers.
- Offers are accepted only if a players if exists agent, else themselves, and their directors.
- If there are multiple offers competing each other, the first offer that is accepted by all parties win cancel out the others.
- Clubs can have at most one director and coach at a time, though their previous occupations are also stored.
- Similarly, a player can only work at single club at a time.
- Director cannot send an offer to their own teams.
- Negative price in offer denotes that the director is expecting money in return and positive price denotes that the director is offering money.
- Clubs cannot play matches against themselves.

7. Implementation Plan

In order to satisfy the design requirements described above, we have divided our project into two parts: frontend, and backend. In frontend, we plan to use HTML, CSS, and Javascript languages and React with ReactStrap(bootstrap for react) and Core UI libraries. Frontend will be statically served (as only html/js/css files) to the users' browsers. In backend, we plan to create Rest API with Django and its Django-Rest framework. We supply data to frontend via the Rest API. We also plan to manage our data with PostgreSQL database in which our SQL queries above designed for.



A REST API is an architectural approach to serve data and some functionalities such as register/login. It works in sync with frontend and allows for more dynamic websites. We use REST(representational state transfer) architectural style. We will present our REST API in JSON(Javascript object notation) format so that it will easily converted to javascript object and be ready to use.

8. Website

Our project information website containing reports and project description is below:

<https://selimfirat.github.io/ciftkale/>

<http://github.com/selimfirat/ciftkale>