Bilkent University

Department of Computer Engineering

Spring 2020 CS353 Project

OnlineScout

Project Design Report

Özge Yaşayan 21704154

Rabia Nur Önal 21702641

Fatih Çakır 21601370

Celal Bayraktar 21501801

The link to this report and the previous report can be found at http://ozgey99.github.io/CS353.

Table of Contents

1. Revised ER Diagram

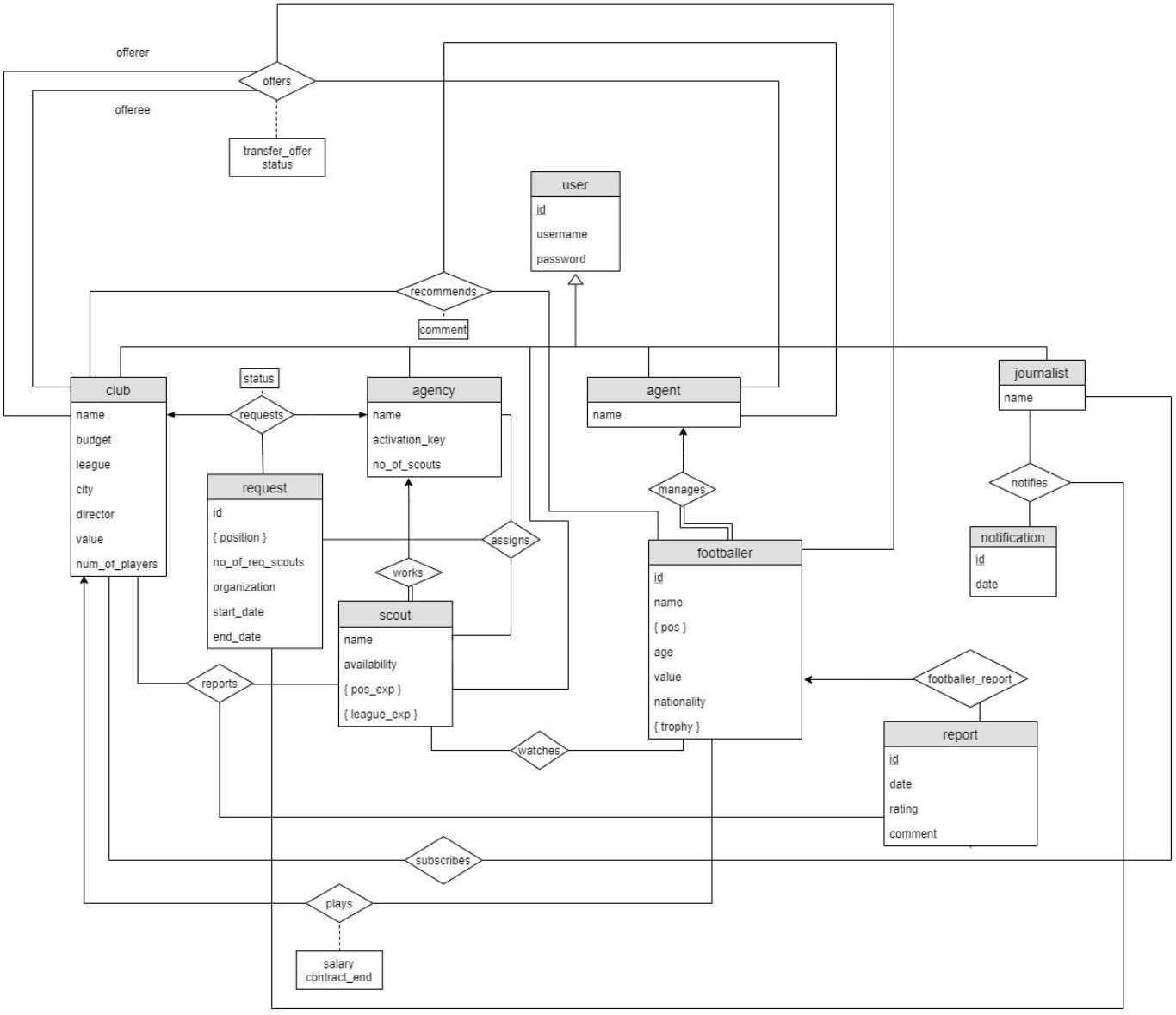

Figure 1: ER diagram

2. Changes Made to the Previous Diagram

- Request, report, notification entities were added.
- Status attribute was added to request entity.
- Recommends relation was added between club, agent, footballer entities.
- Footballer_report relation was added between footballer and report entities.
- Scout was made a user.
- Scout, agent, agency, club, journalist don't have id's anymore, since they are all users.
- Assigns relation is now a ternary relationship with the request entity being tied to it.
- Journalist, notification and request are tied to each other with the notifies relation.
- Offers relation contains the footballer, agent, offering club and the offeree club entities now.
- Requests relation has now different cardinality constraints (1-1-M).

3. Relation Schemas

3.1 User

Relational Model: user (<u>id</u>, username, password)

Functional Dependencies: {(id → username, password)}

Candidate Keys: {(id)}

Primary Key: {(id)}

Normal Form: Since id is the primary key, the relation is in BCNF.

Table Definition:

```
CREATE TABLE ` user` (
  `id` int NOT NULL UNIQUE AUTO_INCREMENT,
  `username` varchar(45) NOT NULL,
  `password` varchar(45) NOT NULL,
  PRIMARY KEY (`id`)
);
```

3.2 Journalist

Relational Model: journalist (<u>id</u>, name)

Functional Dependencies: {(id → name)}

Candidate Keys: {(id)}

Primary Key: {(id)}

Foreign Keys: {(id)} in user table

Normal Form: Since id is the primary key, the relation is in BCNF.

Table Definition:

```
CREATE TABLE `journalist` (
  `id` INT NOT NULL UNIQUE,
  `name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`id`) REFERENCES ` user` (`id`)
);
```

3.3 Agent

Relational Model: agent (id, name)

Functional Dependencies: {(id → name)}

Candidate Keys: {(id)}

Primary Key: {(id)}

Foreign Key: (id) in user table

Normal Form: Since id is the primary key, the relation is in BCNF.

Table Definition:

```
CREATE TABLE `agent` (
  `id` INT NOT NULL UNIQUE,
  `name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`id`) REFERENCES ` user` (`id`)
);
```

3.4 Agency

Relational Model: agency (id, name, activation_key, num_of_scouts)

Functional Dependencies: {(id → name, activation_key, num_of_scouts)}

Candidate Keys: {(id), (activation_key)}

Primary Key: {(id)}

Foreign Keys: {(id)} in user table

Normal Form: Since id is the primary key, the relation is in BCNF.

Table Definition:

```
CREATE TABLE `agency` (
  `id` INT NOT NULL UNIQUE,
  `name` VARCHAR(45) NOT NULL,
  `activation_key` INT NULL,
  `no_of_scouts` INT DEFAULT 0,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`id`) REFERENCES ` user` (`id`)
);
```

3.5 Club

Relational Model: club (id, name, budget, league, city, director, value, num_of_players)

Functional Dependencies: {(id → name, budget, league, city, director, value, num_of_players)}

Candidate Keys: {(id), (director), (name, league), (name, city)}

Primary Key: {(id)}

Foreign Keys: {(id)} in user table

Normal Form: Since id is the primary key, the relation is in BCNF.

Table Definition:

```
CREATE TABLE `club` (
  `id` int NOT NULL UNIQUE,
  `name` varchar(45) NOT NULL,
  `budget` int DEFAULT NULL,
  `league` varchar(45) NOT NULL,
  `city` varchar(45) NOT NULL,
  `director` varchar(45) DEFAULT NULL,
  `value` int DEFAULT NULL,
  `num_of_players` int DEFAULT 0,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`id`) REFERENCES ` user` (`id`)
);
```

3.6 Request

Relational Model: request (id, no_of_req_scouts, organization, start_date, end_date)

Functional Dependencies: {(id → no_of_req_scouts, organization, start_date, end_date)}

Candidate Keys: {(id)}

Primary Key: {(id)}

Foreign Keys: {(id)} in user table

Normal Form: Since id is the primary key, the relation is in BCNF.

Table Definition:

```
CREATE TABLE `request` (
  `id` int NOT NULL UNIQUE AUTO_INCREMENT,
  `no_of_req_scouts` int DEFAULT NULL,
  `organization` varchar(45) DEFAULT NULL,
  `start_date` datetime NOT NULL,
  `end_date` datetime NOT NULL,
  PRIMARY KEY (`id`)
);
```

3.6.1 Request Positions

Relational Model: request_positions (id, position)

Functional Dependencies: {}

Candidate Keys: {(id, position)}

Primary Key: {(id, position)}

Foreign Keys: {(id)} in request table

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `request_positions` (
  `id` int NOT NULL,
  `position` varchar(45) NOT NULL,
  PRIMARY KEY (`id`, `position`),
  FOREIGN KEY (`id`) REFERENCES `request` (`id`)
);
```

3.7 Scout

Relational Model: scout (id, name, availability)

Functional Dependencies: {(id → name, availability)}

Candidate Keys: {(id)}

Primary Key: {(id)}

Foreign Keys: {(id)} in user table

Normal Form: Since id is the primary key, the relation is in BCNF.

Table Definition:

```
CREATE TABLE `scout` (
  `id` INT NOT NULL UNIQUE,
  `name` VARCHAR(45) NOT NULL,
  `availability` TINYINT NOT NULL DEFAULT 1,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`id`) REFERENCES ` user` (`id`));
);
```

3.7.1 Scout Position Experience

Relational Model: scout_position_exp (id, position)

Functional Dependencies: {}

Candidate Keys: {(id, position)}

Primary Key: {(id, position)}

Foreign Keys: {(id)} in scout table

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `scout_position_exp` (
  `id` INT NOT NULL,
  `position` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`, `position`),
  FOREIGN KEY (`id`) REFERENCES `scout` (`id`)
);
```

3.7.2 Scout League Experience

Relational Model: scout_league_exp (id, league)

Functional Dependencies: {}

Candidate Keys: {(id,league)}

Primary Key: {(id, league)}

Foreign Keys: {(id)} in scout table

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `scout_league_exp` (
  `id` INT NOT NULL,
  `league` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`, `league`),
  FOREIGN KEY (`id`) REFERENCES `scout` (`id`)
);
```

3.8 Footballer

Relational Model: footballer (id, name, age, value, nationality)

Functional Dependencies: {(id → name, age, value, nationality)}

Candidate Keys: {(id)}

Primary Key: {(id)}

Normal Form: Since id is the primary key, the relation is in BCNF.

Table Definition:

```
CREATE TABLE `footballer` (
  `id` INT NOT NULL UNIQUE AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `age` INT NULL,
  `value` DOUBLE NULL,
  `nationality` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`)
);
```

3.8.1 Footballer Positions

Relational Model: footballer_positions (id, position)

Functional Dependencies: {}

Candidate Keys: {(id, position)}

Primary Key: {(id, position)}

Foreign Keys: {(id)} in footballer table

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `footballer_positions` (
  `id` INT NOT NULL,
  `position` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`, `position`),
  FOREIGN KEY (`id`) REFERENCES `footballer` (`id`)
);
```

3.8.2 Footballer Trophies

Relational Model: footballer_trophy (id, trophy)

Functional Dependencies: {}

Candidate Keys: {(id,trophy)}

Primary Key: {(id, trophy)}

Foreign Keys: {(id)} in footballer table

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `footballer_trophy` (
  `id` INT NOT NULL,
  `trophy` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`id`, `trophy`),
  FOREIGN KEY (`id`) REFERENCES `footballer` (`id`)
);
```

3.9 Notification

Relational Model: notification (id, date)

Functional Dependencies: {(id → date)}

Candidate Keys: {(id)}

Primary Key: {(id)}

Normal Form: Since id is the primary key, the relation is in BCNF.

Table Definition:

```
CREATE TABLE `notification` (
  `id` INT NOT NULL,
  `date` VARCHAR(15) NULL,
  PRIMARY KEY (`id`)
);
```

3.10 Report

Relational Model: report (id, date, rating, comment)

Functional Dependencies: {(id → date, rating, comment)}

Candidate Keys: {(id)}

Primary Key: {(id)}

Normal Form: Since id is the primary key, the relation is in BCNF.

Table Definition:

```
CREATE TABLE `report` (
  `id` INT NOT NULL,
  `date` VARCHAR(15) NOT NULL,
  `rating` DOUBLE NULL,
  `comment` VARCHAR(45) NULL,
  PRIMARY KEY (`id`)
);
```

3.11 Subscribes

Relational Model: subscribes (journalist_id, club_id)

Functional Dependencies: {}

Candidate Keys: {(journalist_id, club_id)}

Primary Key: {(journalist_id, club_id)}

Foreign Keys: (journalist_id) as id in journalist table, (club_id) as id in club table

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `subscribes` (
  `journalist_id` INT NOT NULL,
  `club_id` INT NOT NULL,
  PRIMARY KEY (`journalist_id`, `club_id`)
  FOREIGN KEY (`journalist_id`) REFERENCES `journalist` (`id`),
  FOREIGN KEY (`club_id`) REFERENCES `club` (`id`)
);
```

3.12 Watches

Relational Model: watches (scout_id, footballer_id)

Functional Dependencies: {}

Candidate Keys: {(scout_id, footballer_id)}

Primary Key: {(scout_id, footballer_id)}

Foreign Keys: (scout_id) as id in scout table, (footballer_id) as id in footballer table

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `watches` (
  `scout_id` INT NOT NULL,
  `footballer_id` INT NOT NULL,
  PRIMARY KEY (`scout_id`, `footballer_id`)
  FOREIGN KEY (`scout_id`) REFERENCES `scout` (`id`),
  FOREIGN KEY (`footballer_id`) REFERENCES `footballer` (`id`)
);
```

3.13 Manages

Relational Model: manages (footballer_id, agent_id)

Functional Dependencies: {}

Candidate Keys: {(footballer_id)}

Primary Key: {(footballer_id)}

Foreign Keys: (agent_id) as id in agent table, (footballer_id) as id in footballer table

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `manages` (
  `footballer_id` INT NOT NULL,
  `agent_id` INT NOT NULL,
  PRIMARY KEY (`footballer_id`)
  FOREIGN KEY (`footballer_id`) REFERENCES `footballer` (`id`),
  FOREIGN KEY (`agent_id`) REFERENCES `agent` (`id`)
);
```

3.14 Footballer Report

Relational Model: footballer_report (report_id, footballer_id)

Functional Dependencies: {}

Candidate Keys: {(report_id)}

Primary Key: {(report_id)}

Foreign Keys: (footballer_id) as **id** in **footballer**, (report_id) as *id* in *report*

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `footballer_report` (
  `report_id` INT NOT NULL,
  `footballer_id` INT NOT NULL,
  PRIMARY KEY (`report_id`)
  FOREIGN KEY (`report_id`) REFERENCES `report` (`id`),
  FOREIGN KEY (`footballer_id`) REFERENCES `footballer` (`id`)
);
```

3.15 Plays

Relational Model: plays (footballer_id, club_id, salary, contract_end)

Functional Dependencies: {(footballer_id → club_id, salary, contract_end)}

Candidate Keys: {(footballer_id)}

Primary Key: {(footballer_id)}

Foreign Keys: (club_id) as id in club, (footballer_id) as id in footballer

Normal Form: Since footballer_id is the primary key, this relation is in BCNF.

Table Definition:

```
CREATE TABLE `plays` (
  `footballer_id` INT NOT NULL,
  `club_id` INT NOT NULL,
  `salary` VARCHAR(15) NULL,
  `contract_end` VARCHAR(15) NULL,
  PRIMARY KEY (`footballer_id`)
  FOREIGN KEY (`footballer_id`) REFERENCES `footballer` (`id`),
  FOREIGN KEY (`club_id`) REFERENCES `club` (`id`)
);
```

3.16 Works

Relational Model: works (scout_id, agency_id)

Functional Dependencies: {scout_id → agency_id }

Candidate Keys: {(scout_id)}

Primary Key: {(scout_id)}

Foreign Keys: (agency_id) as id in agency, (scout_id) as id in scout

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `works` (
  `scout_id` INT NOT NULL,
  `agency_id` INT NOT NULL,
  PRIMARY KEY (`scout_id`)
```

```
  FOREIGN KEY (`scout_id`) REFERENCES `scout` (`id`),
  FOREIGN KEY (`agent_id`) REFERENCES `agent` (`id`)
);
```

3.17 Assigns

Relational Model: assigns (agency_id, scout_id)

Functional Dependencies: {}

Candidate Keys: {(agency_id, scout_id)}

Primary Key: {(agency_id, scout_id)}

Foreign Keys: (agency_id) as id in agency, (scout_id) as id in scout

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `assigns` (
  `agency_id` INT NOT NULL,
  `request_id` INT NOT NULL,
  `scout_id` INT NOT NULL,
  PRIMARY KEY (`agency_id`, `request_id`, `scout_id`)
  FOREIGN KEY (`agency_id`) REFERENCES `agency` (`id`),
  FOREIGN KEY (`request_id`) REFERENCES `request` (`id`),
  FOREIGN KEY (`scout_id`) REFERENCES `scout` (`id`)
);
```

3.18 Requests

Relational Model: requests (request_id, club_id, agency_id, status)

Functional Dependencies: {(request_id → club_id, agency_id, status)}

Candidate Keys: {(request_id)}

Primary Key: {(request_id)}

Foreign Keys: (club_id) as id in club, (agency_id) as id in agency, (request_id) as id in request

Normal Form: request_id makes up the primary key, so this relation is in BCNF.

Table Definition:

```
CREATE TABLE `requests` (
  `request_id` int NOT NULL,
  `club_id` int NOT NULL,
  `agency_id` int NOT NULL,
  `status` varchar(10) NOT NULL DEFAULT ‘pending’,
  PRIMARY KEY (`request_id`),
  FOREIGN KEY (`request_id`) REFERENCES `request` (`id`),
  FOREIGN KEY (`club_id`) REFERENCES `club` (`id`),
  FOREIGN KEY (`agency_id`) REFERENCES `agency` (`id`)
);
```

3.19 Recommends

Relational Model: recommends (footballer_id, agent_id, club_id, comment)

Functional Dependencies: {(footballer_id, agent_id, club_id → comment)}

Candidate Keys: {(footballer_id, agent_id, club_id)}

Primary Key: {(footballer_id, agent_id, club_id)}

Foreign Keys: (footballer_id) as id in footballer, (agent_id) as id in agent, (club_id) as id in club

Normal Form: footballer_id, agent_id, club_id make up the primary key so this relation is in BCNF.

Table Definition:

```
CREATE TABLE `recommends` (
  `agent_id` INT NOT NULL,
  `club_id` INT NOT NULL,
  `footballer_id` INT NOT NULL,
  `comment` VARCHAR(45) NULL,
  PRIMARY KEY (`agent_id`, `club_id`, `footballer_id`)
  FOREIGN KEY (`agent_id`) REFERENCES `agent` (`id`),
  FOREIGN KEY (`club_id`) REFERENCES `club` (`id`),
  FOREIGN KEY (`footballer_id`) REFERENCES `footballer` (`id`)
);
```

3.20 Reports

Relational Model: reports (club_id, scout_id, report_id)

Functional Dependencies: {}

Candidate Keys: {(club_id, scout_id, report_id)}

Primary Key: {(club_id, scout_id, report_id)}

Foreign Keys: (club_id) as id in club, (scout_id) as id in scout, (report_id) as id in report

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `reports` (
  `report_id` INT NOT NULL,
  `scout_id` INT NOT NULL,
  `club_id` INT NOT NULL,
  PRIMARY KEY (`report_id`)
  FOREIGN KEY (`report_id`) REFERENCES `report` (`id`),
  FOREIGN KEY (`scout_id`) REFERENCES `scout` (`id`),
  FOREIGN KEY (`club_id`) REFERENCES `club` (`id`)
);
```

3.21 Notifies

Relational Model: notifies (journalist_id, notification_id, request_id)

Functional Dependencies: {}

Candidate Keys: {(journalist_id, notification_id, request_id)}

Primary Key: {(journalist_id, notification_id, request_id)}

Foreign Keys: (journalist_id) as id in journalist, (notification_id) as id in notification,

(request_id) as id in request

Normal Form: BCNF

Table Definition:

```
CREATE TABLE `notifies` (
  `notification_id` INT NOT NULL,
  `journalist_id` INT NOT NULL,
  `request_id` INT NOT NULL,
  PRIMARY KEY (`notification_id`, `journalist_id`, `request_id`)
  FOREIGN KEY (`notification_id`) REFERENCES `notification` (`id`),
  FOREIGN KEY (`journalist_id`) REFERENCES `journalist` (`id`),
  FOREIGN KEY (`request_id`) REFERENCES `request` (`id`)
```

);

3.22 Offers

Relational Model: offers (offerer_id, offeree_id, footballer_id, agent_id, transfer_offer, status)

Functional Dependencies: {(offerer_id, offeree_id, footballer_id, agent_id, transfer_offer → status)}

Candidate Keys: {(offerer_id, offeree_id, footballer_id, agent_id, transfer_offer)}

Primary Key: {(offerer_id, offeree_id, footballer_id, agent_id, transfer_offer)}

Foreign Keys: (offerer_id) as id in club, (offeree_id) as id in club, (footballer_id) as id in footballer

(agent_id) as id in agent

Normal Form: Since offerer_id, offeree_id, footballer_id, agent_id, transfer_offer make up the primary key, this relation is in BCNF.

Table Definition:

```
CREATE TABLE `offers` (
  `offerer_id` INT NOT NULL,
  `offeree_id` INT NOT NULL,
  `footballer_id` INT NOT NULL,
  `agent_id` INT NOT NULL,
  `transfer_offer` VARCHAR(15) NOT NULL,
  `status` VARCHAR(10) NOT NULL DEFAULT 'pending',
  PRIMARY KEY (`offerer_id`, `offeree_id`, `footballer_id`, `agent_id`, `transfer_offer`)
  FOREIGN KEY (`offerer_id`) REFERENCES `club` (`id`),
  FOREIGN KEY (`offeree_id`) REFERENCES `club` (`id`),
  FOREIGN KEY (`footballer_id`) REFERENCES `footballer` (`id`),
  FOREIGN KEY (`agent_id`) REFERENCES `agent` (`id`)
);
```

4. UI Design and SQL Statements

4.1 Login Page

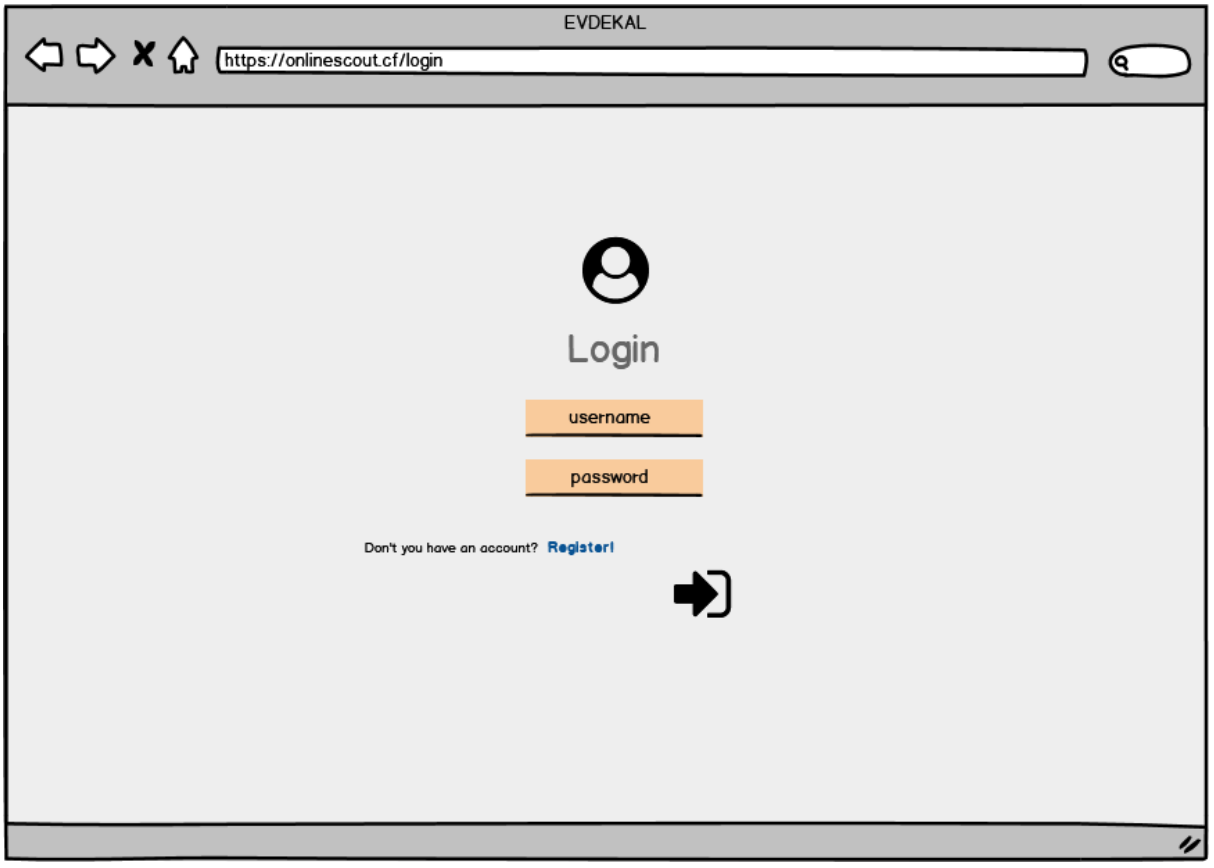

Figure 2: Login Page

```
DELIMITER $$
CREATE PROCEDURE `login`(
    IN name varchar(45),
    IN pw varchar(45),
   OUT u_id int)
BEGIN
            select id
           into u_id
           from user
           where username = name and password = pw;
END$$

DELIMITER ;
```

4.2 Main Sign Up Page

Figure 3: Sign Up Page

4.3 Sign Up as Agency

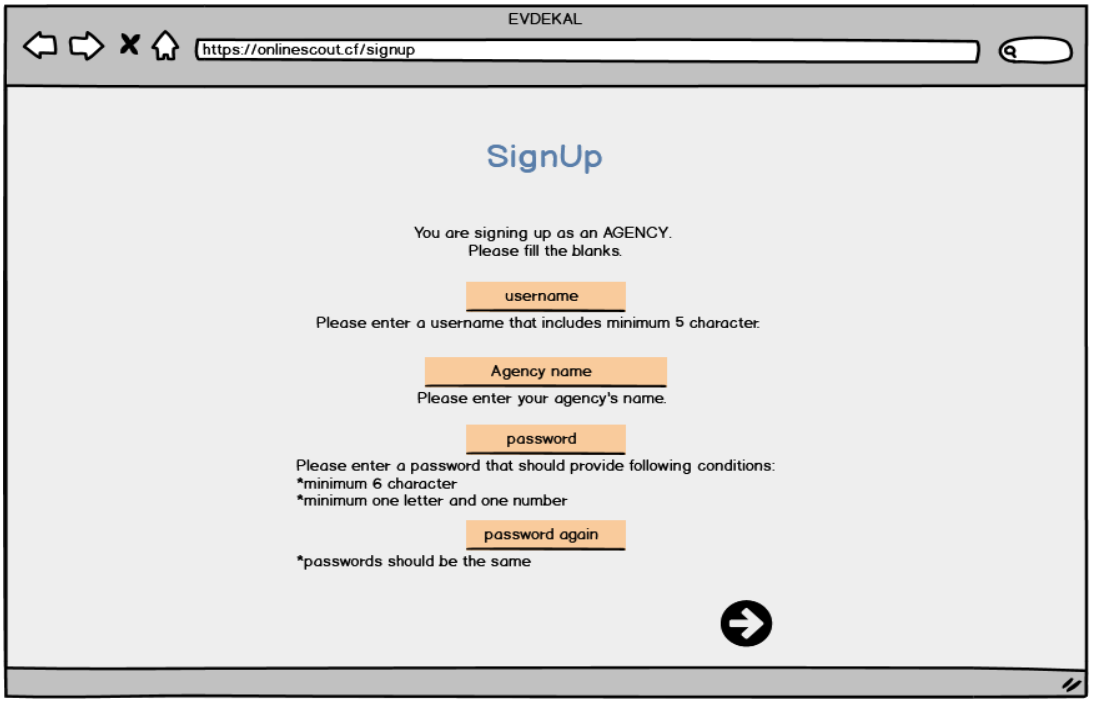

Figure 4: Agency Sign Up Page

```
DELIMITER $$
CREATE PROCEDURE `agencySignUp`(
    IN nick varchar(45),
    IN pw int,
    IN agency_name varchar(45),
   OUT u_id int
)
BEGIN
        insert into user(username, password)
        values(nick, pw);

        select id
        into u_id
        from user
        where id = (select max(id) from user);

        insert into agency(id, name)
        values(u_id, agency_name);
END$$

DELIMITER ;
```

4.4 Sign Up as Agent

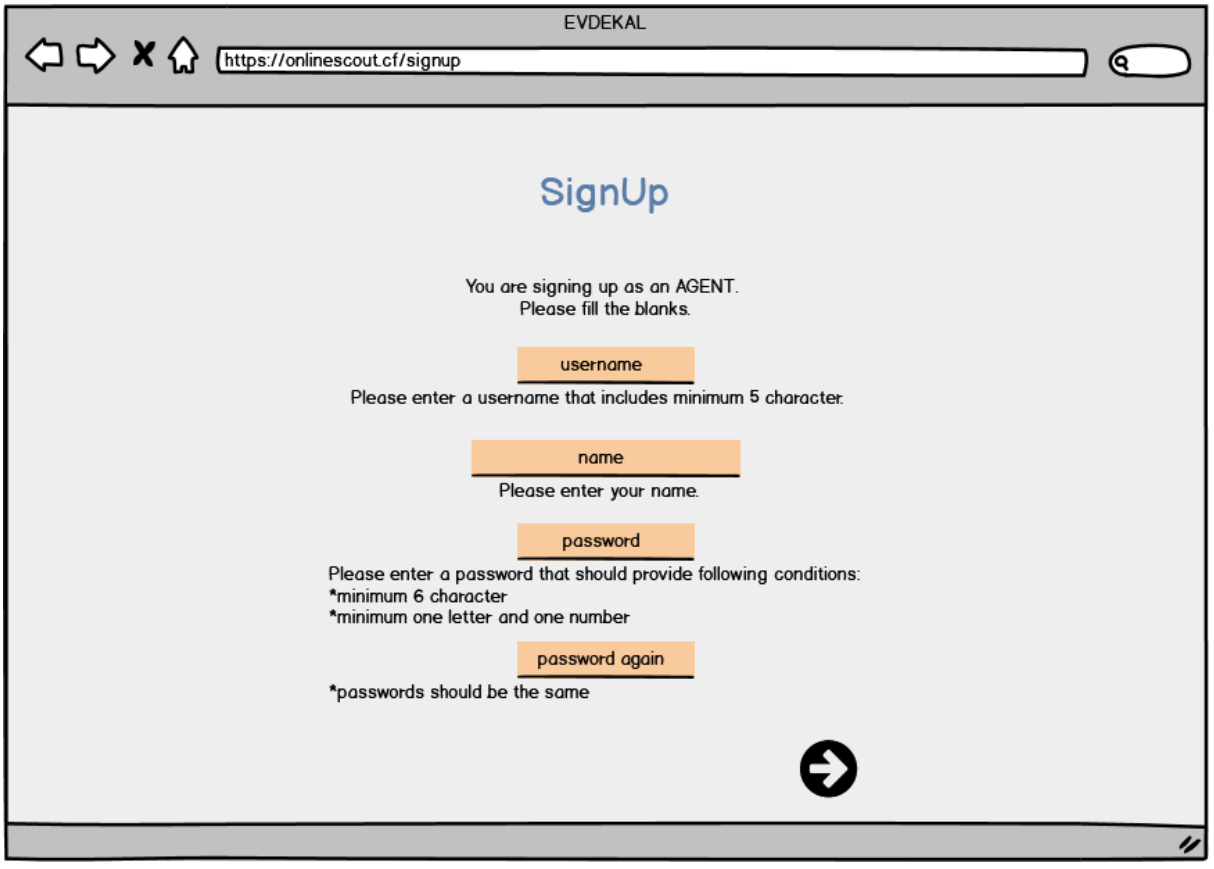

Figure 5: Agent Sign Up Page

```
DELIMITER $$
CREATE PROCEDURE `agentSignUp` (
   IN nick varchar(45),
   IN pw int,
   IN agent_name varchar(45),
   OUT u_id int)
BEGIN
        insert into user(username, password)
        values(nick, pw);

        select id
        into u_id
        from user
        where id = (select max(id) from user);

        insert into agent(id, name)
        values(u_id, agent_name);
END$$

DELIMITER ;
```

4.5 Sign Up as Journalist

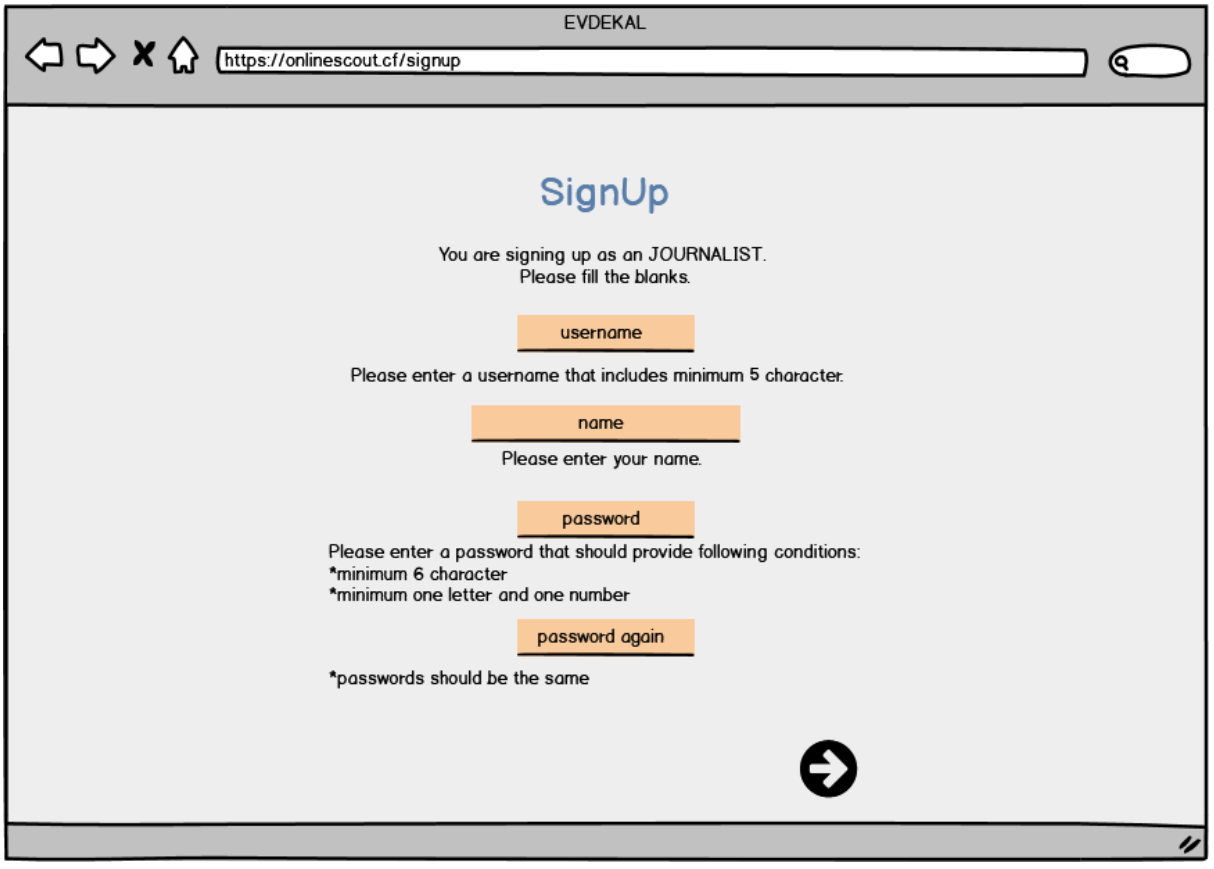

Figure 6: Journalist Sign Up Page

```
DELIMITER $$
CREATE PROCEDURE `journalistSignUp` (
    IN nick varchar(45),
    IN pw int,
    IN journalist_name varchar(45),
   OUT u_id int)
BEGIN
        insert into user(username, password)
        values(nick, pw);

        select id
        into u_id
        from user
        where id = (select max(id) from user);

        insert into journalist(id, name)
        values(u_id, journalist_name);
END$$

DELIMITER ;
```

4.6 Sign Up as Scout

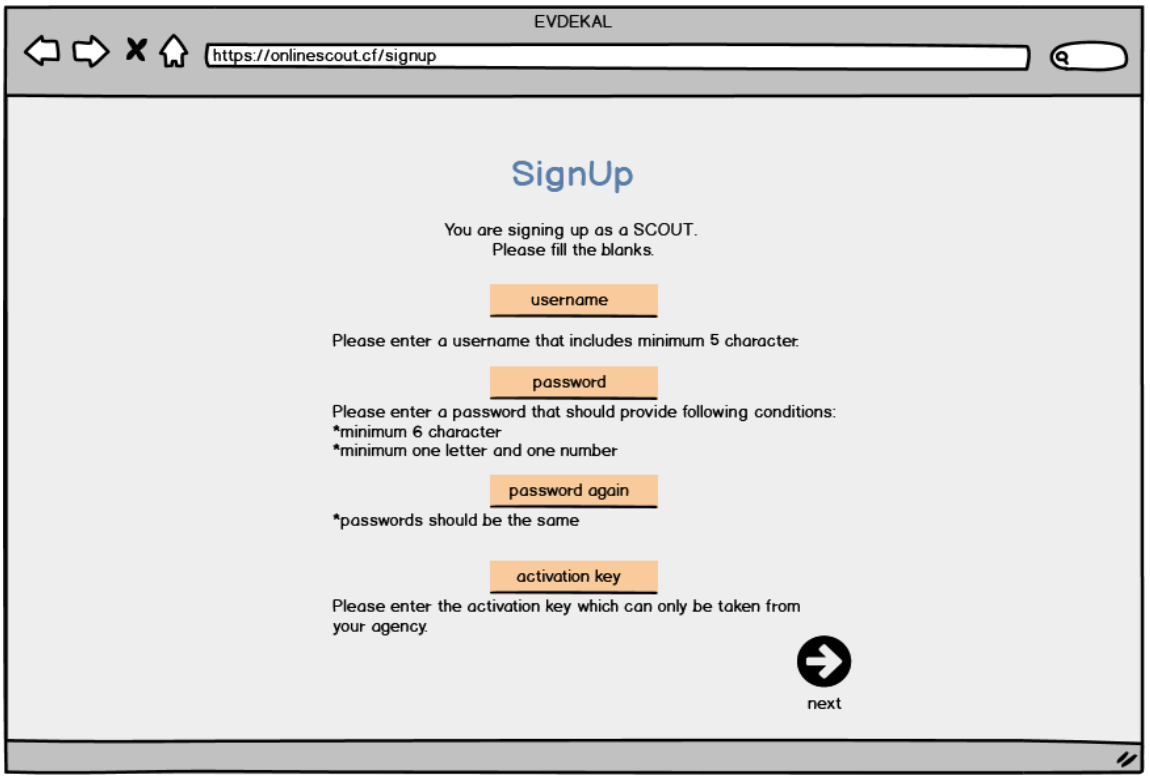

Figure 7.1: Scout Sign Up Page 1

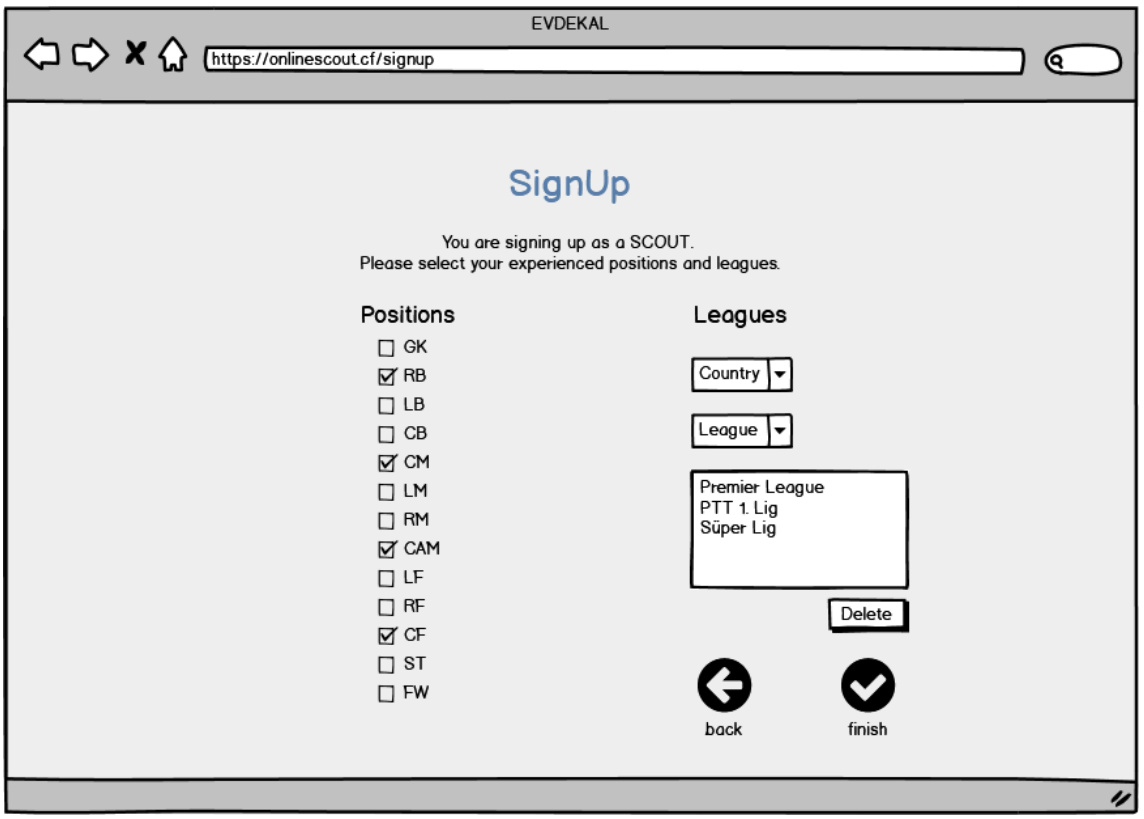

Figure 7.2: Scout Sign Up Page 2

```
DELIMITER $$
CREATE PROCEDURE `scoutSignUp` (
        IN nick varchar(45),
        IN pw int,
        IN scout_name varchar(45),
        IN pos varchar(45),
        IN leags varchar(100),
        IN act_key int,
        OUT u_id int
   )
BEGIN
   if act_key in(select activation_key from agency) then
         insert into user(username, password)
         values(nick, pw);

         select id
         into u_id
         from user
         where id = (select max(id) from user);

        insert into scout(id, name)
         values(u_id, scout_name);
```

```
        set @sql = concat ("INSERT INTO `scout_position_exp`(id, position)
                values(u_id, '', pos , '')");
         prepare stmt from @sql;
         execute stmt;

         set @sql = concat ("INSERT INTO `scout_league_exp`(id, league)
                values(u_id, '', leags , '')");
         prepare stmt from @sql;
         execute stmt;

        insert into works(scout_id, agency_id)
        values(u_id, (select id from agency where activation_key = act_key));

        update agency
        set no_of_scouts = no_of_scouts + 1;
end if;
END$$

DELIMITER ;
```

4.7 Assigning Scouts

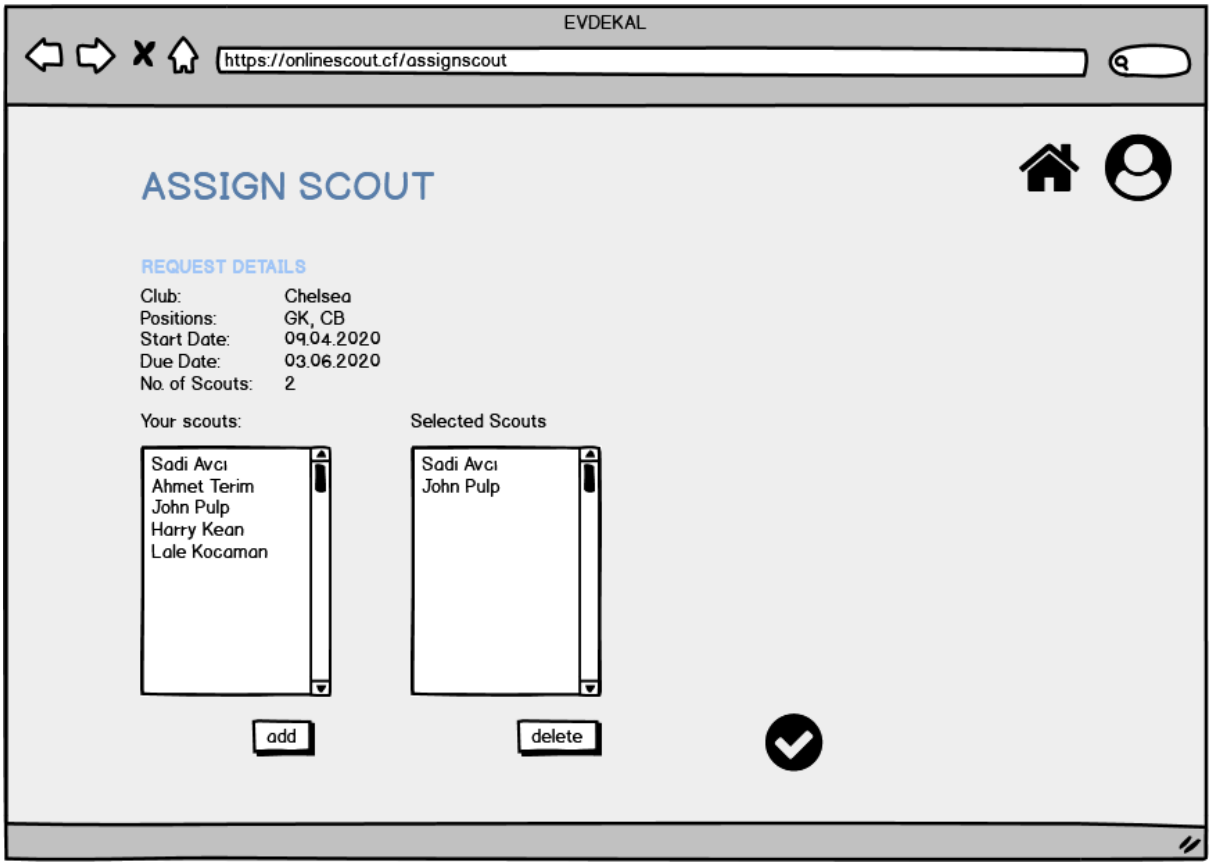

Figure 8: Scout Assigning Page

```
DELIMITER $$
CREATE PROCEDURE `assignScouts` (
   IN req_id int,
   IN ag_id int,
   IN scout_ids varchar(45))
BEGIN
        set @sql = concat ("INSERT INTO `assigns`(request_id, agency_id, scout_id)
              values(req_id, ag_id, ", scout_ids , ")");

        prepare stmt from @sql;
        execute stmt;
END$$

DELIMITER ;
```

4.8 Subscribing to Clubs as a Journalist

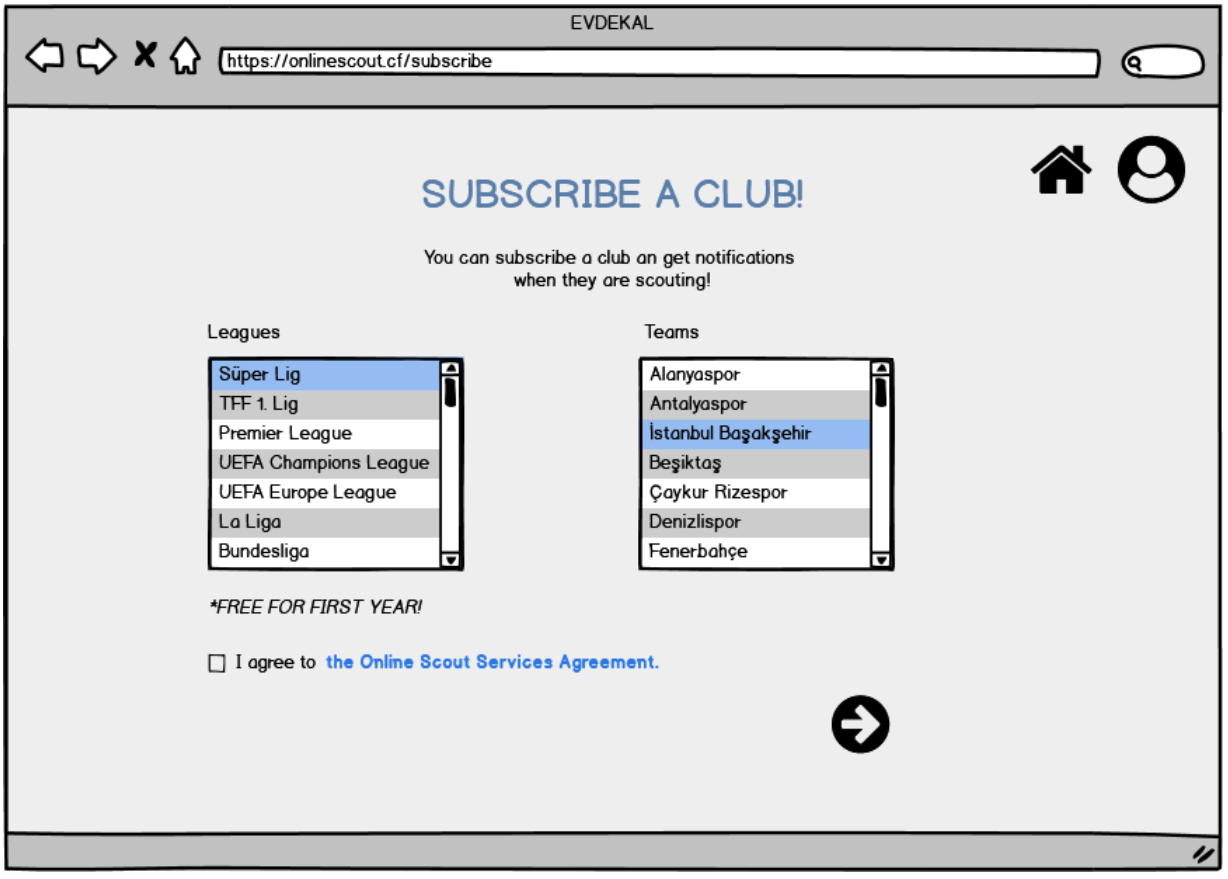

Figure 9: Journalist Subscribing Page

```
DELIMITER $$
CREATE PROCEDURE `subscribeToClub` (
    IN cl_id int,
    IN j_id int
)
BEGIN
        insert into subscribes(club_id, journalist_id)
        values(cl_id, j_id);
END$$

DELIMITER ;
```

4.9 Responding to Requests as an Agency

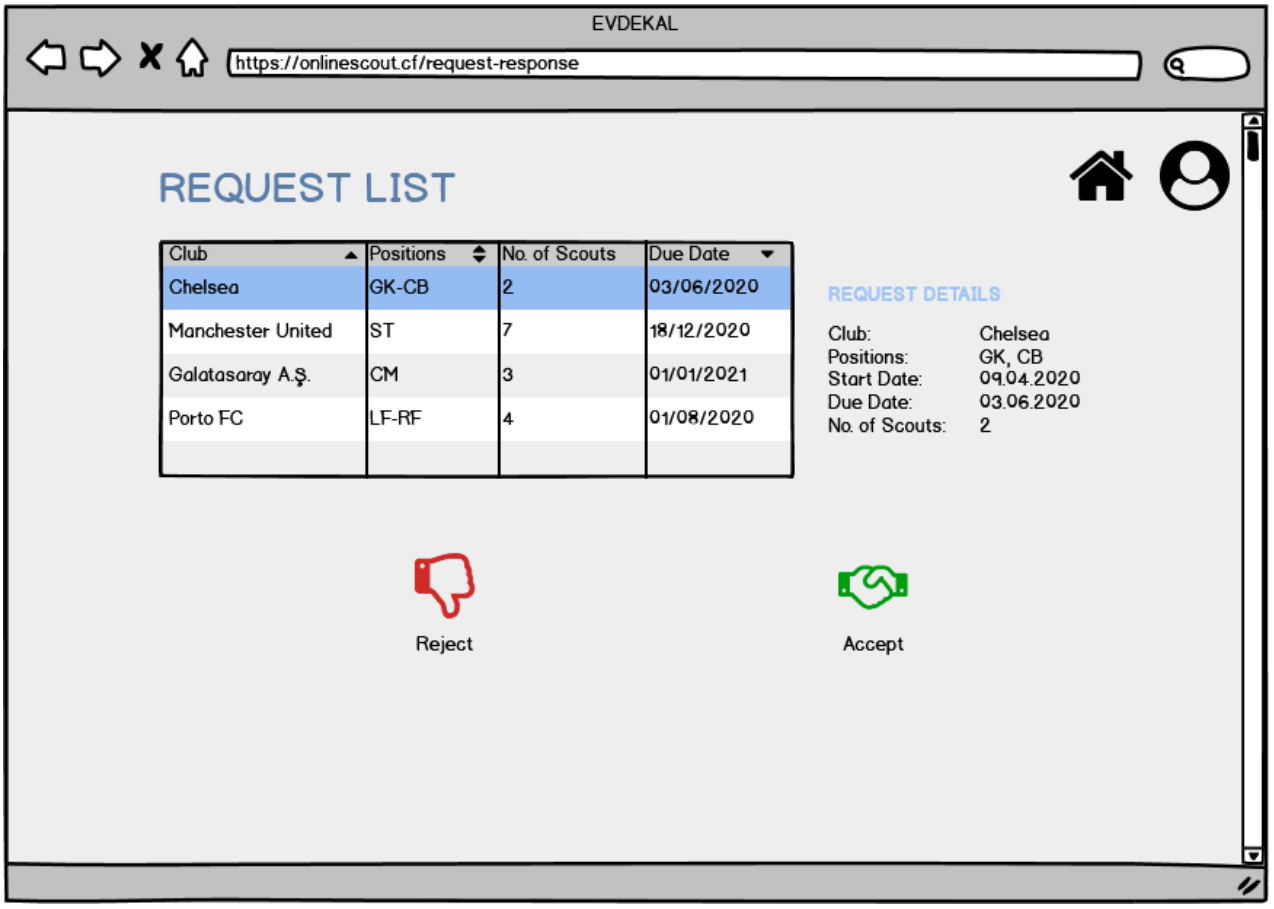

Figure 10: Agency Request List Page

```
DELIMITER $$
CREATE PROCEDURE `respondToReq`(
	IN response varchar(45),
	IN req_id int,
	IN ag_id int)
BEGIN
	update requests
	set status = response
	where request_id = req_id and agency_id = ag_id;
END$$

DELIMITER ;
```

4.10 Requesting Scouts as a Club

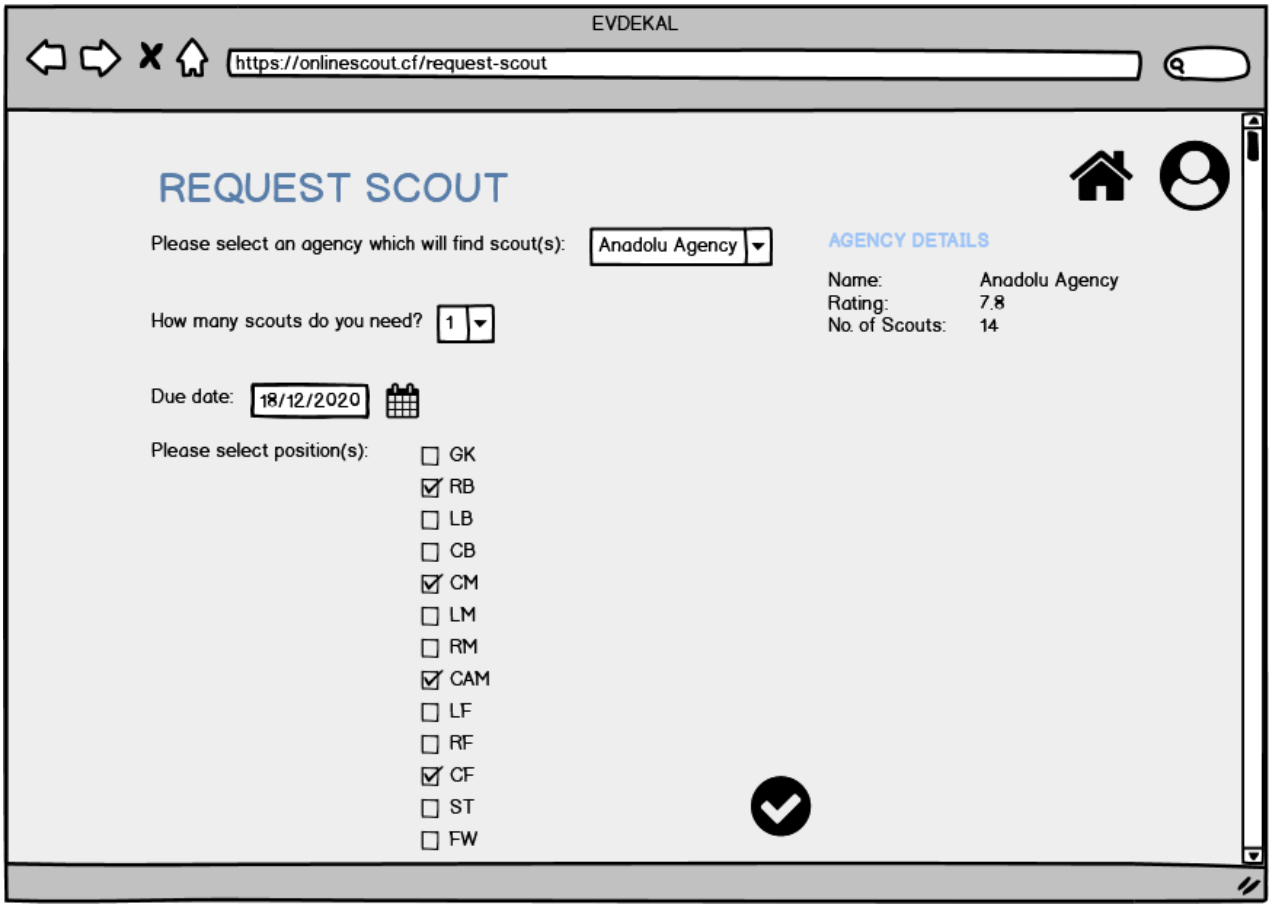

Figure 11: Club Making a Request Page

```
DELIMITER $$
CREATE PROCEDURE `makeRequest` (
        IN no_of_scouts int,
        IN orgztn varchar(45),
        IN start_d datetime,
        IN end_d datetime,
        IN pos varchar(100),
        OUT request_id int)
BEGIN
   insert into request(no_of_req_scouts, organization, start_date, end_date)
values(no_of_scouts, orgztn, start_d, end_d);

        select id
        into request_id
        from request
        where id = (select max(id) from request);
        set @sql = concat ("INSERT INTO `requests_positions`(id, position)
                                        values(request_id, ", pos , ")");

        prepare stmt from @sql;
        execute stmt;
END$$

DELIMITER ;


DELIMITER $$
CREATE PROCEDURE `sendRequest` (
        IN req_id int,
        IN cl_id int,
        IN ag_id int
   )
BEGIN
   insert into requests(request_id, club_id, agency_id)
   values(req_id, cl_id, ag_id);
END$$

DELIMITER ;
```

4.11 Seeing Notifications as a Journalist

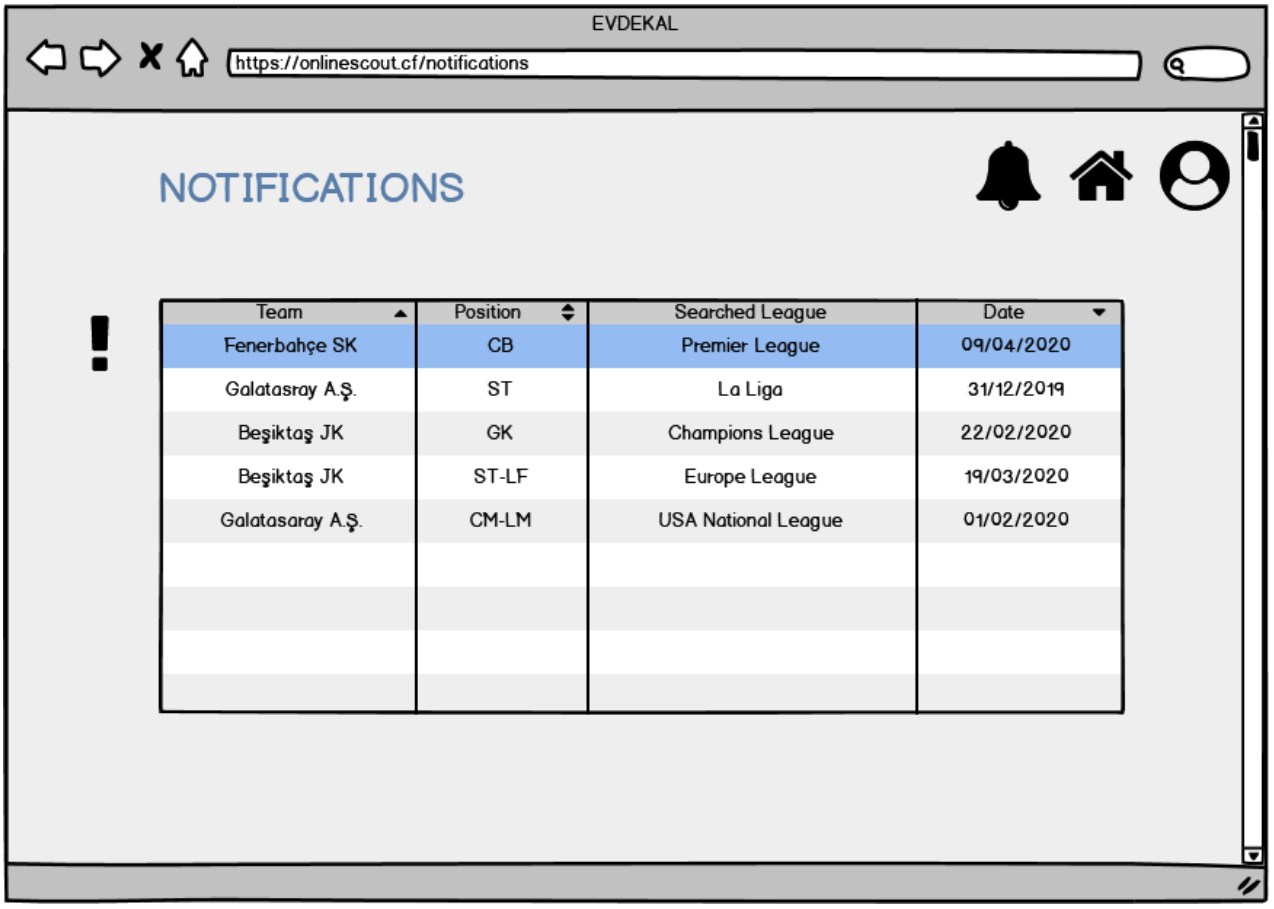

Figure 12: Journalist Notifications Page

```
DELIMITER $$

drop trigger notify_journalist;
CREATE TRIGGER notify_journalist
AFTER INSERT
ON requests FOR EACH ROW
BEGIN
  insert into notification(date)
  values(curdate());

  insert into notifies(notification_id, journalist_id, request_id)
  select (select max(id) from notification), journalist.id, NEW.request_id
  from journalist, subscribes
  where subscribes.journalist_id = journalist.id and subscribes.club_id = NEW.club_id;
END$$

DELIMITER ;
```

4.12 Recommending a Player as an Agent

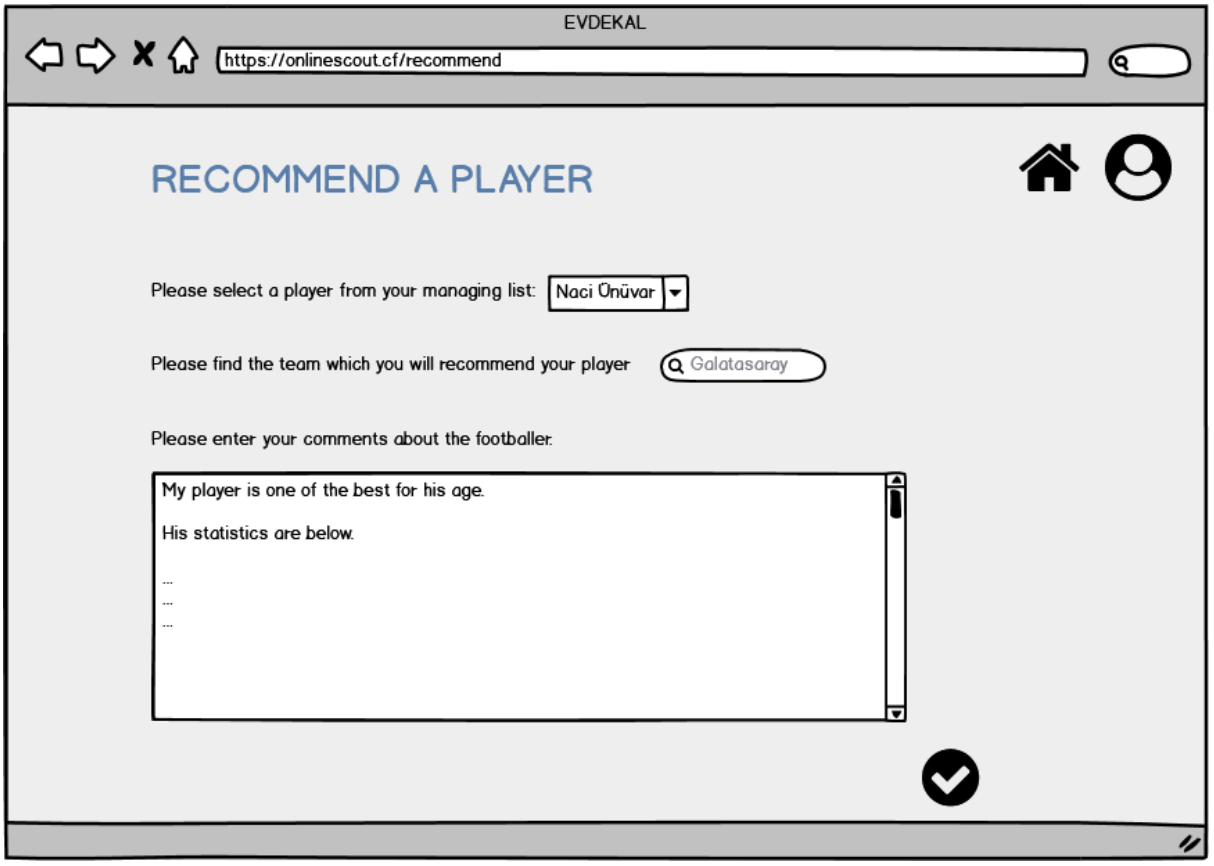

Figure 13: Agent - Recommend a Footballer Page

```
DELIMITER $$
CREATE PROCEDURE `recommendFootballer`(
    IN ag_id int,
    IN pl_id int,
    IN cl_id int,
    IN cmmt varchar(100))
BEGIN
        insert into recommends(agent_id, club_id, footballer_id, comment)
        values(ag_id, cl_id, pl_id, cmmt);
END$$

DELIMITER ;
```

4.13 Submitting a Report as a Scout

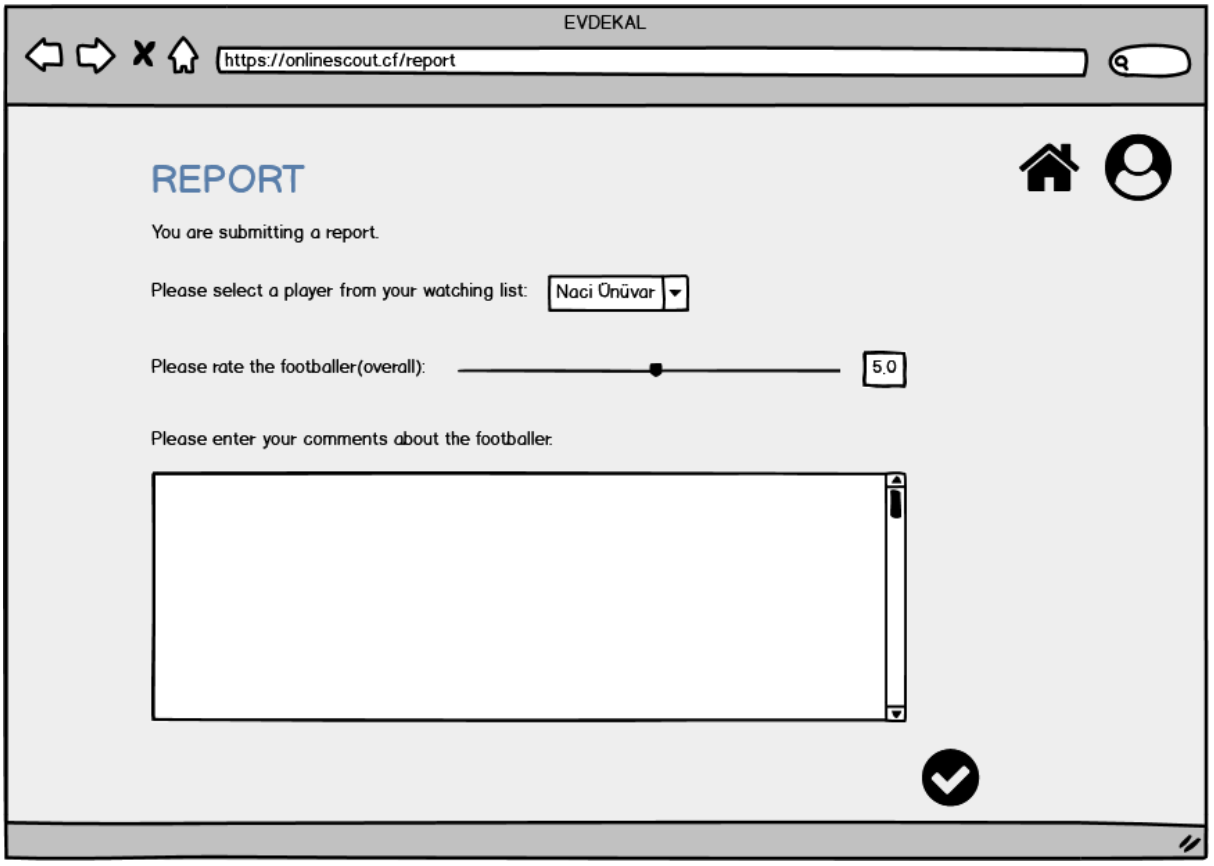

Figure 14: Scout - Report for a Footballer Page

```
DELIMITER $$
CREATE PROCEDURE `submitReport` (
        IN sc_id int,
        IN cl_id int,
        IN pl_id int,
        IN cmmt varchar(45),
        IN rtg double,
        IN dt datetime,
       OUT r_id int)
BEGIN
          insert into report(date, rating, comment)
          values(dt, rtg, cmmt);

          select id
          into r_id
          from report
          where r_id = (select max(id) from report);

          insert into footballer_report(footballer_id, report_id)
```

```
        values(pl_id, r_id);

        insert into reports(scout_id, club_id, report_id)
        values(sc_id, cl_id, r_id);
END$$

DELIMITER ;
```

4.14 Seeing Reports as a Club

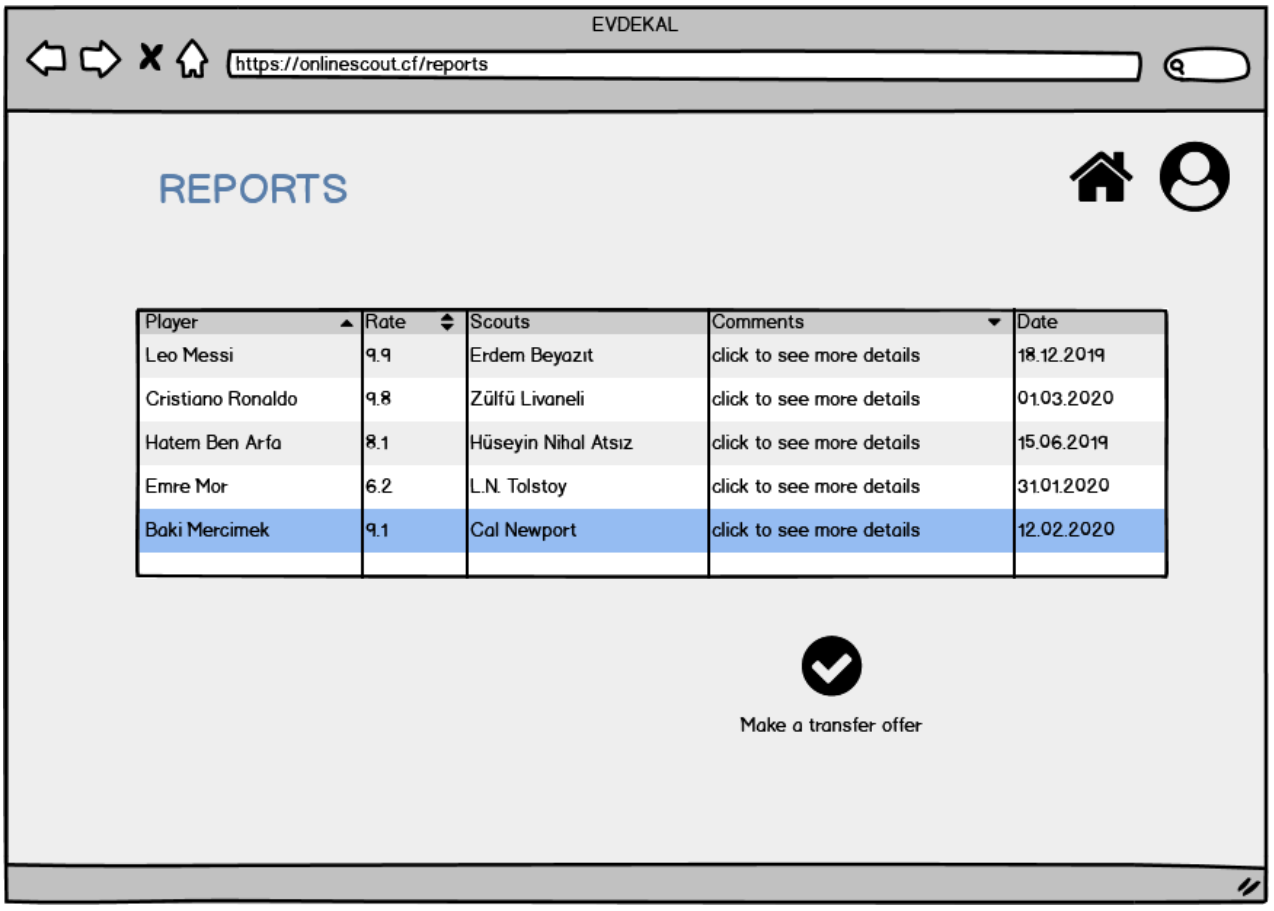

Figure 15: Report List for a Club Page

```
DELIMITER $$
CREATE PROCEDURE `seeReportsForClub` (
   IN cl_id int)
BEGIN
        select footballer.name, scout.name, report.rating, report.comment, report.date
        from report, footballer_report, footballer, scout
        where report.scout_id = scout.id and report.id = footballer_report.report_id
        and footballer_report.footballer_id = footballer.id and report.club_id = cl_id;
END$$
DELIMITER ;
```

4.15 Seeing Requests as an Agency

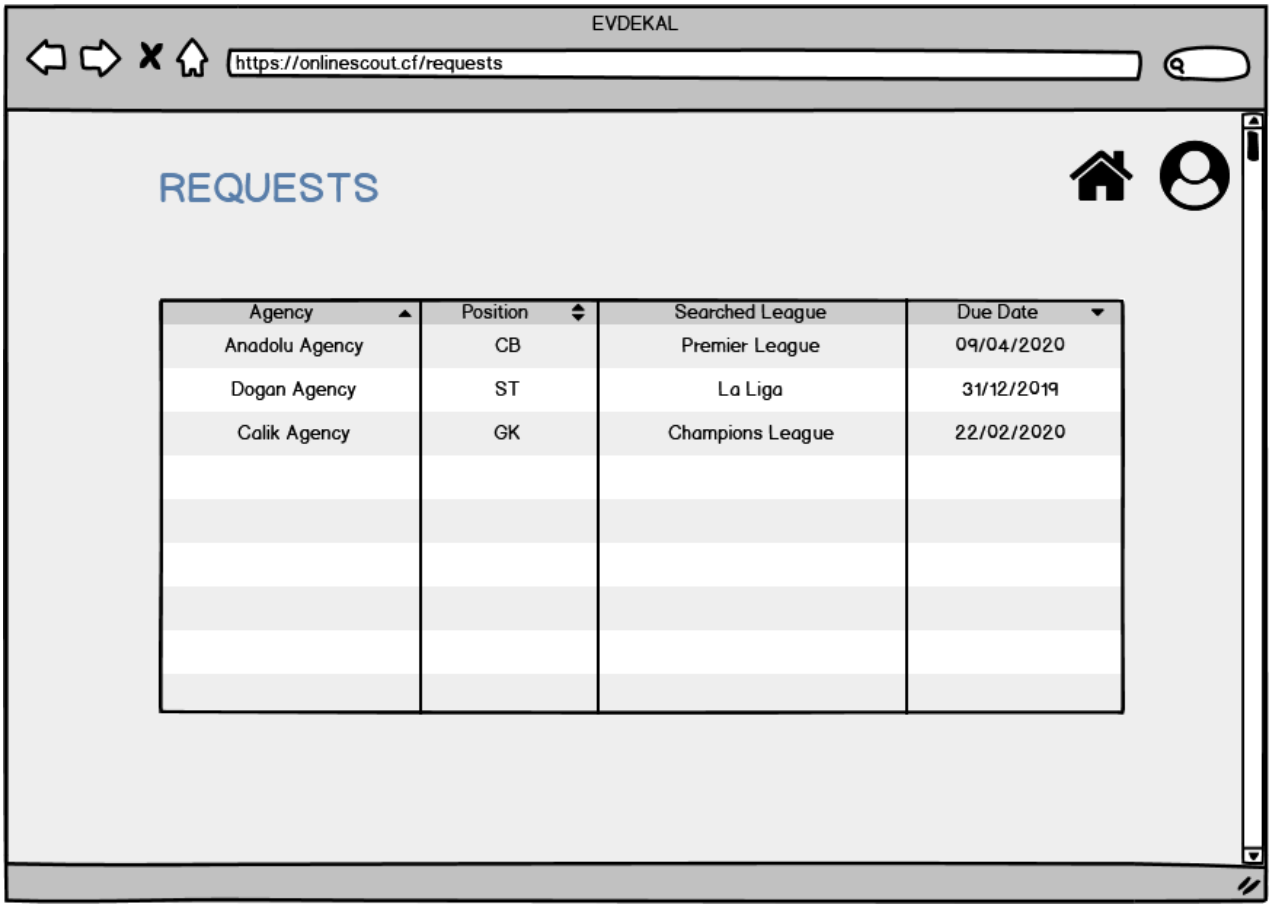

Figure 16: Requests for an Agency Page

```
DELIMITER $$
CREATE PROCEDURE `seeRequests` (
   IN ag_id int)
BEGIN
        select club.name, no_of_req_scouts, end_date, start_date, organization, position,
                status
        from request, clubs, requests, request_positions
        where requests.agency_id = ag_id and requests.request_id = request.id
        and club.id = requests.club_id and request_positions.id = request.id;
END$$

DELIMITER ;
```

4.16 Seeing Tasks as a Scout

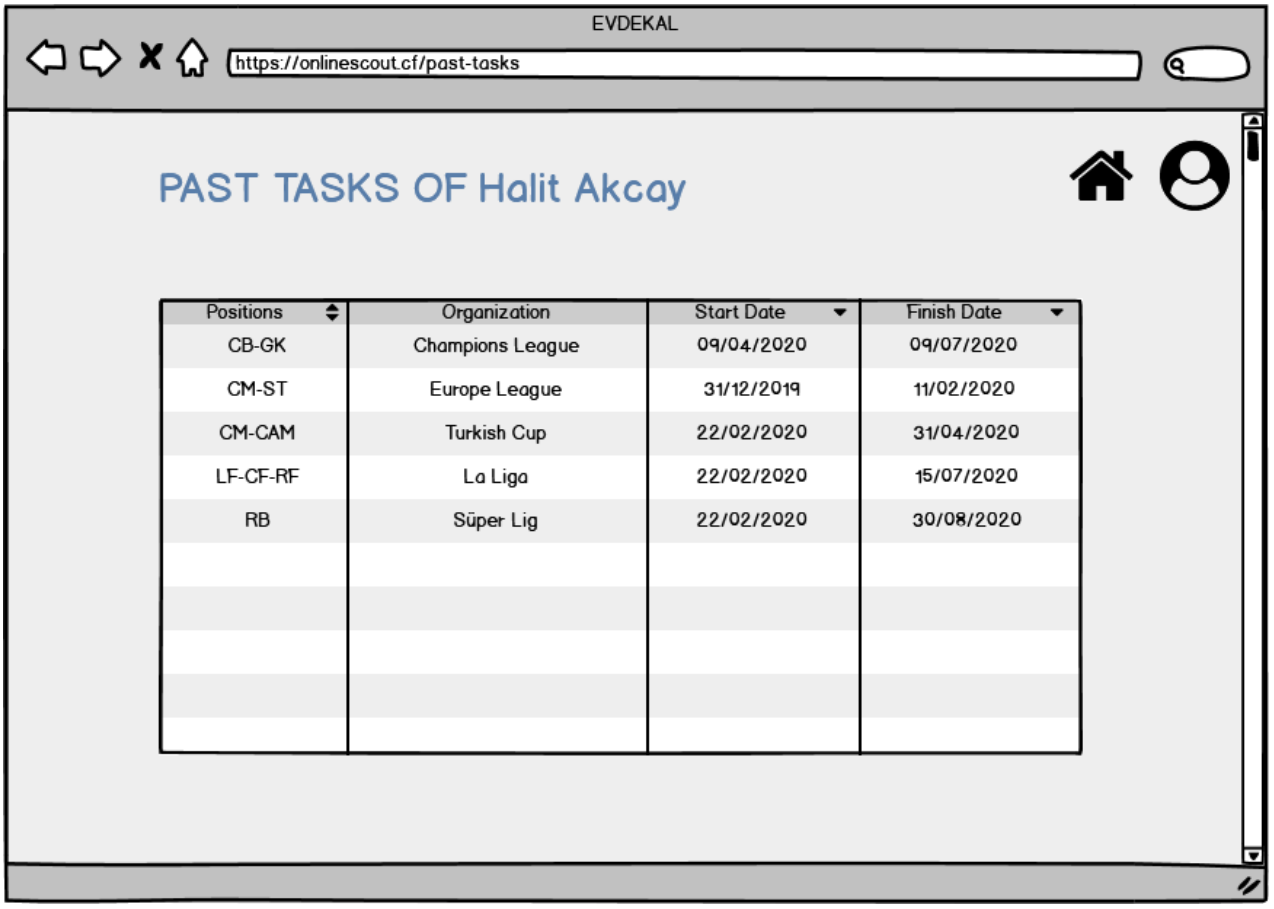

Figure 17: Tasks of a Scout Page

```
DELIMITER $$
CREATE PROCEDURE `seeTasks` (
   IN sc_id int)
BEGIN
        select position, organization, end_date, start_date
        from request_positions, request, assigns
        where assigns.scout_id = sc_id and request_positions.id = request.id
        and assigns.request_id = request.id;
END$$

DELIMITER ;
```

4.17 Making a Transfer Offer as a Club

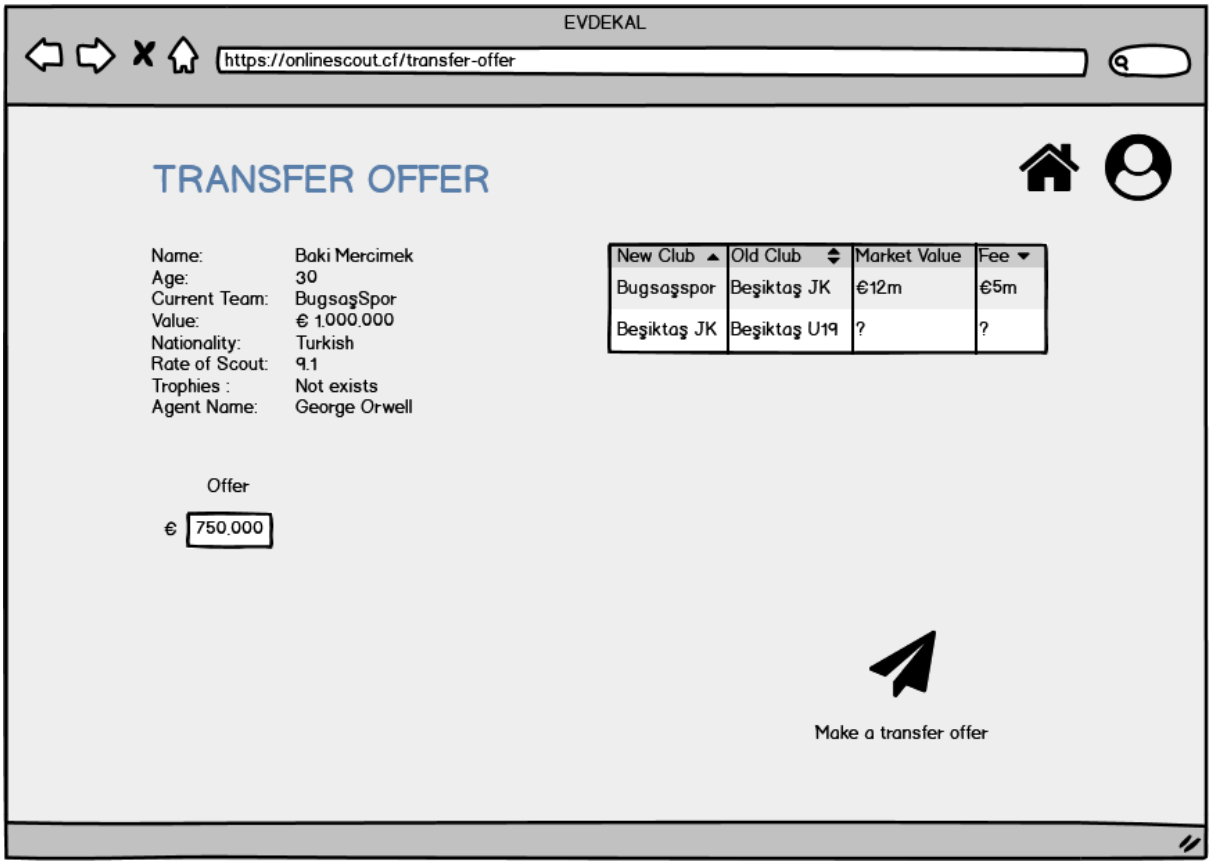

Figure 18: Transfer Offering Page

```
DELIMITER $$
CREATE PROCEDURE `getPrevTransfers` (
   IN pl_id int)
BEGIN
        select c1.name as offeree, c2.name as offerer, transfer_offer
        from club c1, club c2, offers
        where offers.offeree_id = c1.id and offers.offerer_id = c2.id
        and offers.status = 'approved' and offers.footballer_id=pl_id;
END$$

DELIMITER ;
```

```
DELIMITER $$
CREATE PROCEDURE `makeOffer` (
        IN ofr_id int,
        IN ofe_id int,
        IN ag_id int,
        IN pl_id int,
        IN offer int)
BEGIN
        insert into offers(agent_id, offeree_id, offerer_id, transfer_offer, footballer_id)
        values(ag_id, ofe_id, ofr_id, offer, pl_id);
END$$

DELIMITER ;
```

4.18 Responding to a Transfer Offer

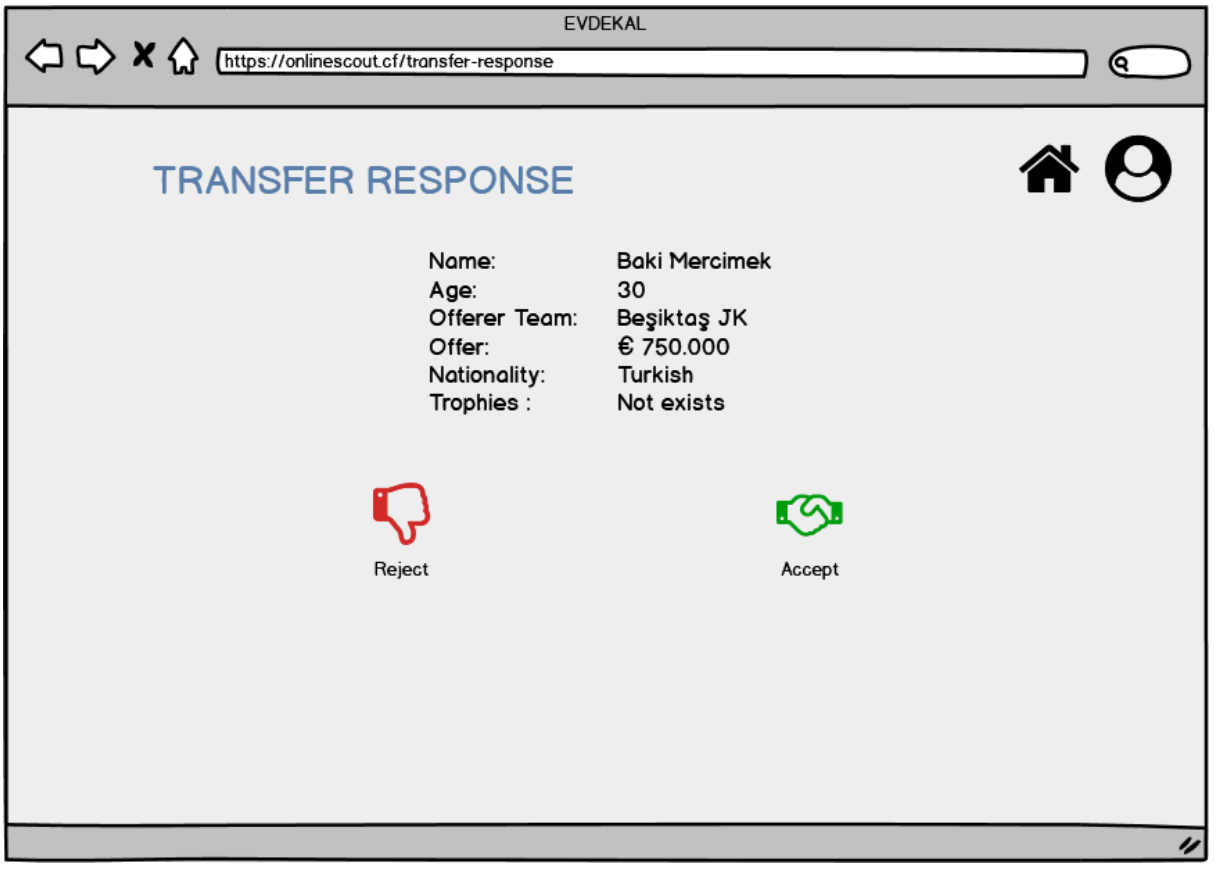

Figure 19: Transfer Response Page

```
DELIMITER $$
CREATE PROCEDURE `respondToOffer`(
	IN response varchar(45),
	IN ofr_id int,
	IN ofe_id int,
	IN ag_id int,
	IN pl_id int,
	IN offer int)
BEGIN

	update offers
	set status = response
	where offerer_id = ofr_id and offeree_id = ofe_id and agent_id = ag_id
	and transfer_offer = offer and footballer_id = pl_id;
END$$

DELIMITER ;
```

5. References

[1] Balsamiq Wireframes 4. [Online] Available: https://balsamiq.com Accessed:10.04.2020