CS353 - Database Management
Spring 2020
Project Final Report

Name of the Project :
Scouting Platform For Football Clubs

Group No : 10

Group Members
Asena Şahin 21502354
Betim Doğan 21502164
Safa Alperen Oruç 21502209
Sera Fırıncıoğlu 21401803

Website URL : https://serafirincioglu.github.io/CS353_DatabaseSystems/#reports
Course Instructor : Uğur Güdükbay

Contents

Description 3

Final E/R 4

Final List of Tables 5

Implementation Details 11

Advanced Database Features and Preliminary Results (Actual Outputs) 12

User Manual 14

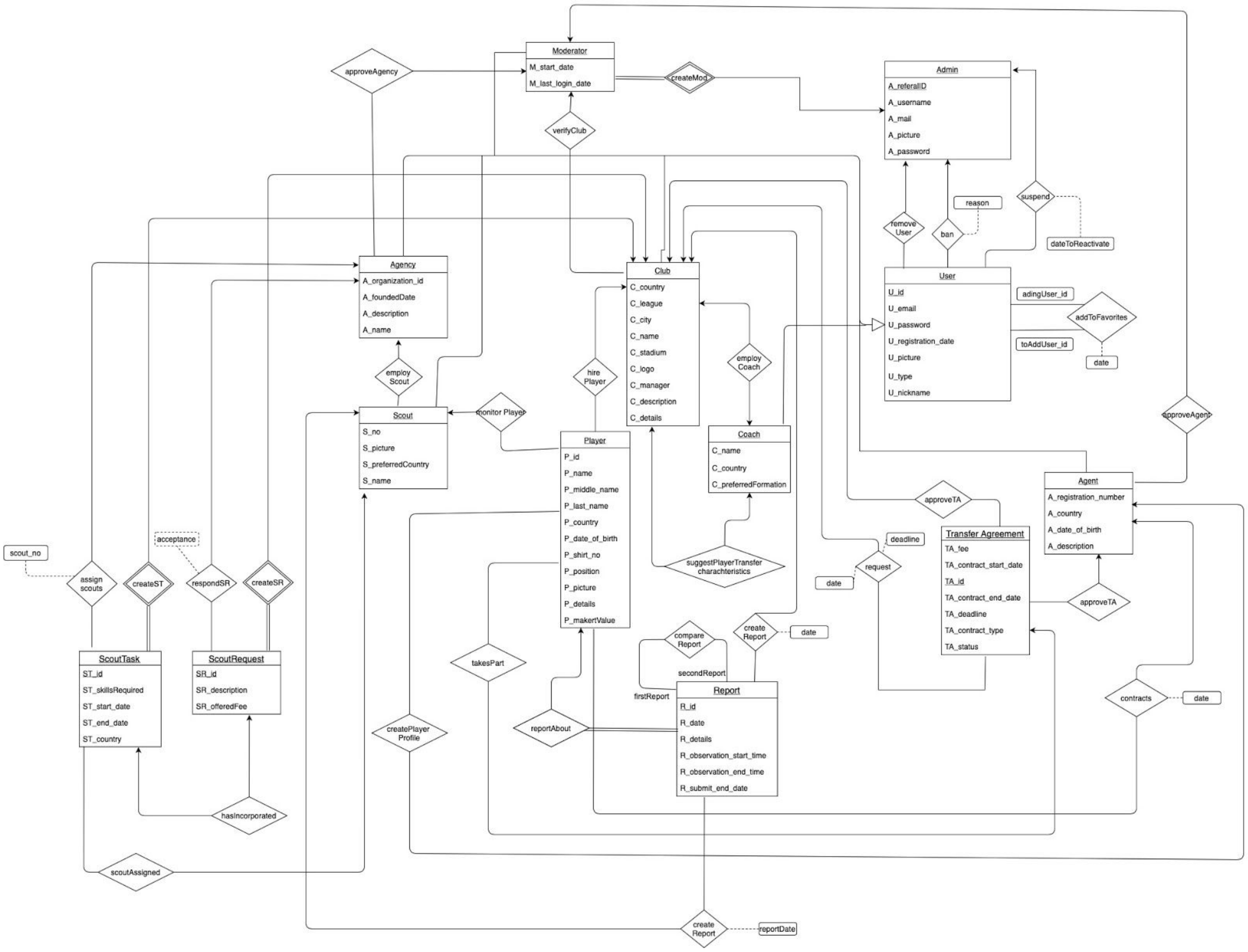
1. Description

Our project is Scouting Platform for Football Clubs which is designed to help Football Clubs to search better football players for their team with the help of agencies. Aim of the project is creating an user-friendly platform for its all users.

In our platform, there are three types of users. Football Club, Agency and Scout. Every user can sign up to the platform by entering their names, emails and then creating their password. Every user has different functionalities in the platform. Our platform also keeps detailed information about the players such as their country, position, market value, transfers.

Basically the working principle of our platform is as follows : Football clubs can search for many different agencies with their specific information such as how many scouts do they have and then select one of them. Agencies have rights to accept or reject the requests taken from football clubs. In case of acceptance, football clubs can create a specific task for their intended request which includes such as which position they are looking for. Later on, the Agency assigns its scouts for this specific task and asks them to create an evaluation report for the football players. The reports that are created by the scouts are sent to the football clubs to compare and evaluate. If clubs decide to make a transfer of a player, it can send a request for transfer to the agent and footballer's club. This request is evaluated by the agency and footballer's club. If both are agreed on the transfer offer, a request can be done.

2. Final E/R



Link for the ER diagram :
https://serafirincioglu.github.io/CS353_DatabaseSystems/CS353_ERDiagram.pdf

3. Final List of Tables

The following gives a view of all the table schemas for the entities used in our ER.

User

Relational Model

User(U_id, U_email, U_password, U_registration_date, U_picture, U_type, U_nickname)
Primary Key U_id

Moderator

Relational Model

Moderator(U_id,A_referralID, M_start_date, M_last_login_date)
Foreign Key U_id references User(U_id)
Foreign Key A_referralID references Admin(A_referralID)

Agency

Relational Model

Agency(U_id, A_organization_id, A_foundedDate, A_description, A_name)
Foreign Key U_id references User(U_id)

Scout

Relational Model

Scout(U_id, S_no, S_picture, S_preferredCountry, S_name)
Foreign Key U_id references User(U_id)

Agent

Relational Model

Agent(U_id, A_registration_number, A_country, A_date_of_birth, A_description)
Foreign Key U_id references User(U_id)

Player

Relational Model

Player(P_id, U_id, P_name, P_middle_name, P_last_name, P_country, P_shirtno, P_position, P_picture, P_marketValue, P_details)
Foreign Key U_id references Agent(U_id)
Primary Key P_id

Coach

Relational Model
Coach(U_id, C_country, C_preferedFormation)
Foreign Key U_id references User(U_id)

Admin

Relational Model
Admin(A_referalID, A_username,A_password, A_accountNo, A_picture)

Transfer Agreement

Relational Model
TransferAgreement(TA_id, TA_fee, TA_contract_start_date, TA_contract_end_date, TA_contract_type, TA_status, TA_deadline)
Primary Key TA_id

Club

Relational Model
User(U_id, C_country, C_league, C_city, C_name, C_stadium, C_logo, C_manager, C_description, C_details)
Foreign Key U_id references User(U_id)

ScoutRequest

Relational Model
ScoutRequest(U_id,SR_id, SR_description, SR_offeredFee)
Primary Key SR_id
Foreign Key U_id references Club(U_id)

ScoutTask

Relational Model

ScoutTask(U_id,ST_id, ST_skillsRequired, ST_country, ST_start_date, ST_end_date)
Primary Key ST_id,U_id
Foreign Key U_id references Club(U_id)

Report

Relational Model
Report(P_id,R_id, R_details, R_observation_start_time, R_observation_end_time, R_submit_end_date)
Primary Key R_id
Foreign Key P_id references Player(P_id)

approveAgency

Relational Model
approveAgency(A_id, M_id, approval_date)
Foreign Key A_id references Agency(U_id)
Foreign Key M_id references Moderator(U_id)

assignScouts

Relational Model
assignScouts(U_id, ST_id,C_id, M_start_date,scout_no)
Foreign Key U_id references Agency(U_id)
Foreign Key ST_id references ScoutTask(ST_id)
Foreign Key C_id references ScoutTask(U_id)

respondSR

Relational Model
respondSR(U_id, SR_id,U_id,acceptance)
Foreign Key U_id references Club(U_id)
Foreign Key SR_id references ScoutRequest(SR_id)
Foreign Key U_id references ScoutRequest(U_id)

hasIncorporated

Relational Model
hasIncorportated(U_id,ST_id, SR_id,C_id)

Foreign Key SR_id references ScoutRequest(SR_id)
Foreign Key ST_id references ScoutTask(ST_id)
Foreign Key U_id references ScoutTask(U_id)
Foreign Key C_id references ScoutRequest(U_id)

scoutAssigned

Relational Model
scoutAssigned(U_id,ST_id,S_id)
Foreign Key S_id references Scout(U_id)
Foreign Key ST_id references ScoutTask(ST_id)
Foreign Key U_id references ScoutTask(U_id)

createReport

Relational Model
createReport(U_id,R_id,P_id, reportDate)
Foreign Key U_id references Club(U_id)
Foreign Key R_id references Report(R_id)
Foreign Key P_id references Report(P_id)

compareReport

Relational Model
compareReport(firstReport,secondReport,firstPlayer,secondPlayer)
Foreign Key firstReport references Report (R_id)
Foreign Key secondReport references Report (R_id)
Foreign Key firstPlayer references Report (P_id)
Foreign Key secondPlayer references Report (P_id)
Primary Key: firstReport,secondReport,firstPlayer,secondPlayer

employScout

Relational Model
employScout(A_id,S_id,date)
Foreign Key A_id references Agency(U_id)
Foreign Key S_id references Scout(U_id)

hirePlayer

Relational Model

hirePlayer(U_id,P_id,date)
Foreign Key U_id references Club(U_id)
Foreign Key P_id references Player(P_id)

employCoach

Relational Model

employCoach(U_id,C_id,date)
Foreign Key U_id references Club(U_id)
Foreign Key C_id references Coach(U_id)

monitorPlayer

Relational Model

monitorPlayer(U_id,P_id,A_id,date)
Foreign Key U_id references Scout(U_id)
Foreign Key P_id references Player(P_id)
Foreign Key A_id references Agent(U_id)

suggestPCharacteristics

Relational Model

suggestPCharacteristics(U_id,C_id,date)
Foreign Key U_id references Club(U_id)
Foreign Key C_id references Coach(U_id)

approveTA

Relational Model

approveTA(U_id,TA_id,date)
Foreign Key U_id references Club(U_id)

approveAgent

Relational Model

approveAgent(M_id,A_referalID,A_id,date)

Foreign Key M_id references Moderator(U_id)

Foreign Key A_referalID references Admin(A_referalID)

Foreign Key A_id references Agent(U_id)

ban

Relational Model

ban(A_id,U_id,date)

Foreign key (U_id) references User(U_id)

Foreign key (A_id) references Admin(A_referralID)

removeUser

Relational Model

removeUser(A_id,U_id,date)

Foreign key (U_id) references User(U_id)

Foreign key (A_id) references Admin(A_referralID)

suspendUser

Relational Model

suspendUser(A_id,U_id,reactivation_date)

Foreign key (U_id) references User(U_id)

Foreign key (A_id) references Admin(A_referralID)

addToFavorites

Relational Model

addToFavorites(U_id,U_fav_id,_date)

Foreign key (U_id) references User(U_id)

Foreign key (U_id) references User(U_fav_id)

4. Implementation Details

We decided to interpret this project as an interactive website that behaves as a dashboard display for the action each user type on the system. In order to make this display we chose php as the programming language for the project as it provides a good system in terms of the access of global and session variables which for this project are crucial for the needed information to run the queries on the database. We also chose PHP because it is relatively easy to run SQL queries and to interpret results as arrays of information. Our Database is MySQL as it provides for all the features that we have studied in this course. We used HTML, CSS and Javascript in small pieces to build the GUI. We made use of open source ready jquery structures to style the tables and display the data.The main problem in this project was having to adapt to a completely new programming language and to spend the time to build the GUI structure that fitted our queries. Having to make all the Setup through a shared team environment we had to synchronise all of the interaction with one another and we made use of XAMPP to make sure that the setup of PhpMyAdmin for the database management, SQL and Apache Servers was all same through all the devices we had.The contribution and distribution of the workload in the group proceeds as follows:

Sera Fırıncıoğlu : Wrote the SQL queries for insertion, update, modification from the database, the HTML code in Bootstrap for the website(usersEntries, agentTransfer, agentListPlayers, signup, addPlayer), the PHP code in the HTML files for data retrieval from the database, the JavaScript code in signup.

Betim Doğan : Wrote the HTML code in Bootstrap for the website (agentDash, clubDash, moderatorDash, agent_homePG, agentListPlayers), the SQL code in PHP for creation and insertion of some tables, the PHP code in the HTML files for data retrieval from the database and the SQL queries for insertion, update, modification from the database.

Asena Şahin : Wrote the SQL queries for insertion, update, modification from the database. Wrote the HTML code in Bootstrap for the website (signup, index, usersEntries, agentTransfer, club_homepg). Wrote the SQL code in php for creation and insertion of some tables. Wrote the PHP code in the HTML files for data retrieval from the database.

Safa Alperen Oruç : Wrote the HTML code in Bootstrap for the website(clubAllPlayers,clubListPlayers,clubTransferReq,agentListTransfer,agentTransfer), SQL code in php for creation and insertion of some tables,PHP code in the HTML files for data retrieval from the database. Wrote the SQL queries for insertion, update, modification from the database.

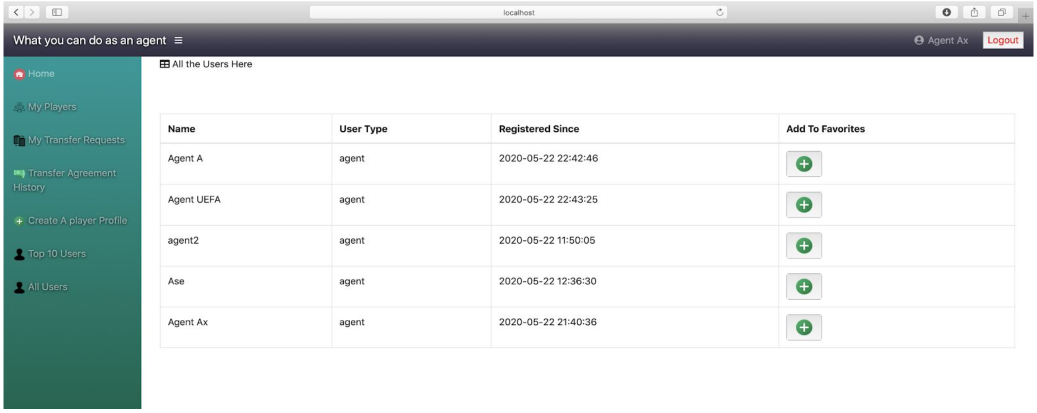
5. Advanced Database Features and Preliminary Results (Actual Outputs)

View :

```
select `seradb`.`addtofavorites`.`U_id` AS `U_id`,count(*) AS `counter` from
`seradb`.`addtofavorites` group by `seradb`.`addtofavorites`.`U_id`
```

The view is used to make it easy to have the count of users that have made the user account as their favorite. This helps to create a ranking system of the users present in the database.

UI : Agent Account, Top 10 Users



All the Users Here

Name	User Type	Registered Since	Add To Favorites
Agent A	agent	2020-05-22 22:42:46	⊕
Agent UEFA	agent	2020-05-22 22:43:25	⊕
agent2	agent	2020-05-22 11:50:05	⊕
Ase	agent	2020-05-22 12:36:30	⊕
Agent Ax	agent	2020-05-22 21:40:36	⊕

AgentTop10List :

```
"SELECT *
   FROM User NATURAL JOIN favoritesCount
   WHERE U_id in (
   SELECT S1.U_id
   FROM favoritesCount AS S1
   WHERE (SELECT COUNT(S2.U_id)
   FROM favoritesCount AS S2
   WHERE S2.counter > S1.counter) <= 9 )";
```

Trigger :

```
BEGIN
   IF new.TA_status="approved" THEN
      INSERT INTO hirePlayer (`U_id`, `P_id`)
                SELECT b.cid,a.Pid
                FROM takesPart a JOIN clubRequest b
                WHERE a.Ta_id=b.taid AND a.Ta_id=new.Ta_id;
   END IF;
END
```

We used an index on player name as it facilitates player search by name. Since the name attribute of the player is text we used a full text spatial index.

6. User Manual

- ❏ Users can create an account by clicking the sign up button and then writing his/her name, email and password to complete the process if she/he does not have any account.
- ❏ Users can log in to the system by entering its emails and password after signing up.
- ❏ Football clubs can search for Agencies by clicking the request scout. Agencies with their name and number of scouts can be seen.
- ❏ Football clubs can create tasks for the scouts that they can specify the details about the football players after they choose the scout agency.
- ❏ Football clubs can create new task by selecting the number of scouts, position and country in create a new task page.
- ❏ Agencies can see its requests by clicking the tasks. Later on the agency can accept or reject with a checkmark which every request has.
- ❏ Football clubs can see whether their tasks are accepted or not which they have sent to the Agency by clicking my tasks.
- ❏ Football clubs can list all the information about their previous task by clicking my tasks.
- ❏ Agencies can list all the requests that they have by clicking the tasks to reject or deny. Accepted tasks can be seen by clicking the accepted tasks.
- ❏ In the accepted tasks screen, Agencies can assign scouts to the selected request by clicking the assign scouts button. After in the assign scout page, the Agency can select the beginning and end of the dates with the desired number of scouts.
- ❏ Scouts can see their current task by clicking my current tasks and previous tasks by clicking my previous tasks.
- ❏ Scouts can start the process of commenting and rating the players by clicking the select players in my current tasks page.
- ❏ Scouts will be directed to the select player page when they hit the select player button and they can put a checkmark to the players to create form. After selection they can create form for the players by clicking create form for selected player.
- ❏ In the create report page Scouts can comment and rate the player and then send the report to the club by clicking the send to club button.
- ❏ Football clubs can evaluate the reports by clicking the Scout Reports.

- ❏ Football clubs can make a transfer offer by clicking the make a transfer offer in their main page. In the directed after clubs can select the players and send requests to the agency and the club of requested players.
- ❏ Football clubs can see the status of their transfer requests status by clicking the transfer status.
- ❏ Football clubs can see the transfer offers that they have taken.
- ❏ Football clubs can see the detailed previous transfer information by clicking footballer transfers.