Bilkent University

CS 353

Database Systems

Project Final Report

Group 14

Şekip Kaan EKİN 21401795 Section 1

Ege Darçın 21302460 Section 1

Can Demirel 21401521 Section 1

Eren Aslantürk 21401306 Section 2

1. Project Description

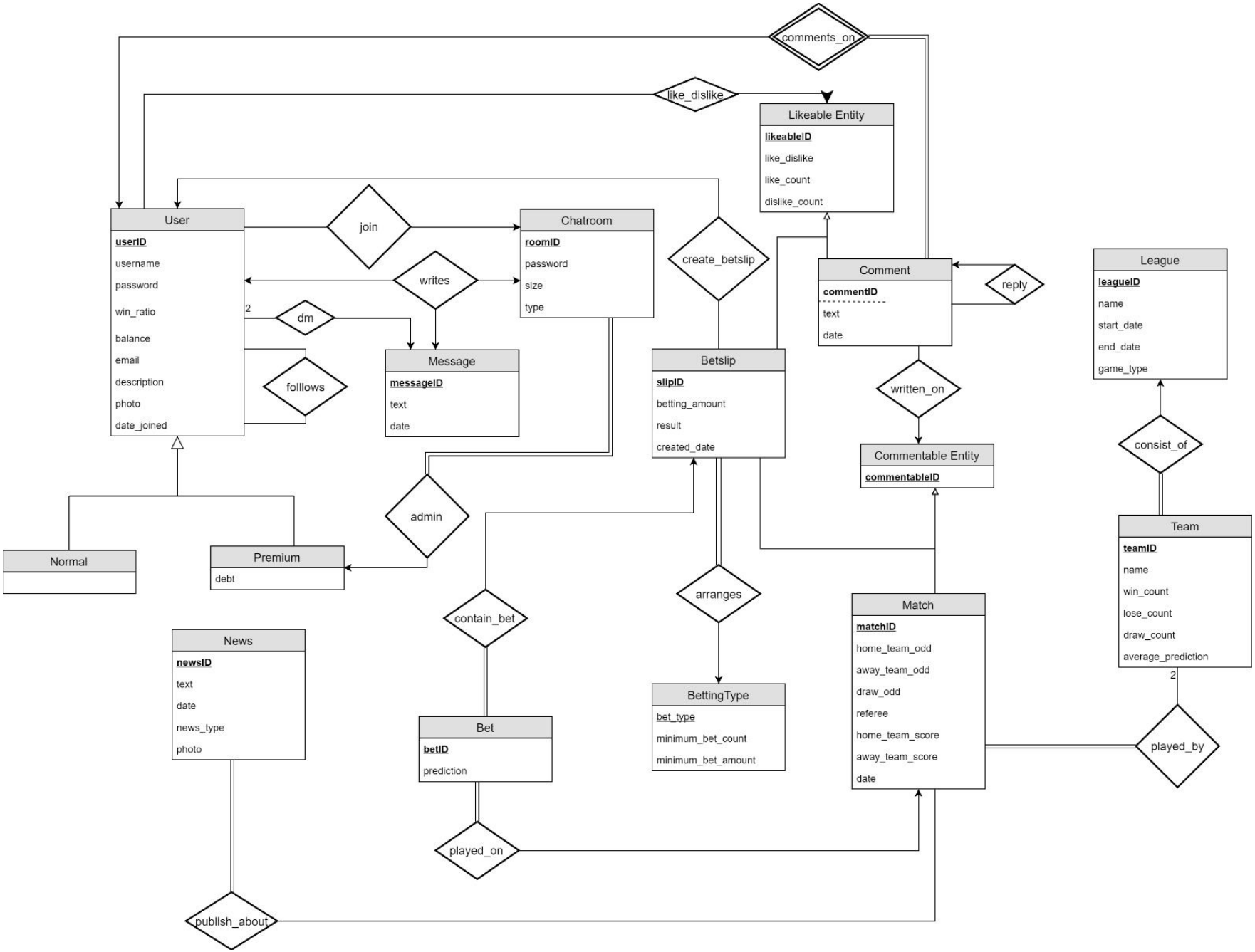
BetBall Database System is a web-based application designed to display and modify the standings of various leagues and tournaments around the world. The system features information about leagues, matches, statistics and bets. It also provides a social platform where users can interact with each other in public and private rooms, private messages and comments. It also displays information about user's monthly report, most winning users and league ranking. Users have win ratio as well as clubs. On their profile they can set a description and photo. They can also see their balances of their accounts. Users have option to become premium users by paying a fee. Premium users can create their own chat rooms. Chat Rooms can have different sizes and may or may not have passwords. They can manage this rooms by giving a title or inviting their friends to their chatroom. Joined users can chat in that room and share their opinions.

Our project realizes a database management application for football and basketball. Regular and Premium Users can review, comment, set bets inside the database. The system will suggest the most rated bets, most winning bets and most winning users in homepage. The news part in homepage will include the match dates and news and images about the sport clubs and leagues. Users can like/dislike betslips or comments. Also they can comment on matches, betslips as well as comments. Users can view the standing matches in leagues and cups by visiting bets page. Team and League information will be provided in bets page. Users can see the win and lose rates of the teams , the start and end date of the matches.

Bets have predictions which can be played on matches. They have different odds for home, away and draw. Betslips consists of different bets which are multiplied on betslip. Betslips can have minimum bet count and minimum bet amount.

2. Revised ER Diagram

- The relation between the user, chatroom and message turned to a ternary relation.
- To avoid misunderstandings, “sends” changed to “dm”.
- “Comments_on” table deleted since weak relations do not require tables.
- Like and dislike counts are held in “like_count” relation, not inside each entity.
- To correct the SQL codes, Comment and Betslip entities now are in a IS-A relation with “likeable_entity” and Betslip and Match entities are in a IS-A relation with “commentable_entity”.


comments_on
like_dislike
Likeable Entity
likeableID
like_dislike
like_count
dislike_count
User
userID
username
password
win_ratio
balance
email
description
photo
date_joined
join
Chatroom
roomID
password
size
type
writes
dm
2
folllows
Message
messageID
text
date
create_betslip
Comment
commentID
text
date
reply
League
leagueID
name
start_date
end_date
game_type
Betslip
slipID
betting_amount
result
created_date
written_on
Commentable Entity
commentableID
consist_of
Team
teamID
name
win_count
lose_count
draw_count
average_prediction
2
Normal
Premium
debt
admin
contain_bet
arranges
Match
matchID
home_team_odd
away_team_odd
draw_odd
referee
home_team_score
away_team_score
date
played_by
News
newsID
text
date
news_type
photo
Bet
betID
prediction
BettingType
bet_type
minimum_bet_count
minimum_bet_amount
played_on
publish_about

2.1 Entity Sets

2.1.1 User

- Relational Model:

user(<u>userID</u>, username, password, win_ratio, balance, email, description, photo, day_joined)

2.1.2 Normal User

- Relational Model:

normal_user(<u>userID</u>)

FK: userID references User

2.1.3 Premium User

- Relational Model:

premium_user(<u>userID</u>, debt)

FK: userID references User

2.1.4 Chatroom

- Relational Model:

chatroom(<u>roomID</u>, password, size, type)

2.1.5 Message

- Relational Model:

message(<u>messageID</u>, text, date)

2.1.6 Likeable Entity

- Relational Model:

likeable_entity(<u>likeableID</u>)

2.1.7 Commentable Entity

- Relational Model:

commentable_entity(commentableID)

2.1.8 Betslip

- Relational Model:

bet_slip(slipID, likeableID, commentableID, betting_amount, result, created_date)

FK: likeableID references likeableEntity

FK: commentableID references commentableEntity

2.1.9 Bet

- Relational Model:

Bet(betID, prediction)

2.1.10 Betting Type

- Relational Model:

betting_type(bet_type, minimum_bet_count, minimum_bet_amount)

2.1.11 Comment

- Relational Model:

comment(userID, commentID, likeableID, text, date)

FK: userID references User

FK: likeableID reference likeableEntity

2.1.12 News

- Relational Model:

news(newsID, text, date, news_type, photo)

2.1.13 Match

- Relational Model:

match(matchID, commentableID, home_team_odd, away_team_odd, draw_odd, referee, home_team_score, away_team_score, date)

FK: commentableID references commentableEntity

2.1.14 Team

- Relational Model:

team(teamID, name, win_count, lose_count, draw_count, average_prediction)

2.1.15 League

- Relational Model:

league(leagueID, name, start_date, end_date, game_type)

2.2 Relation Sets

2.2.1 Join

- Relational Model:

join(<u>userID</u>, roomID)

FK: userID references User
roomID references Chatroom

2.2.2 Direct Message

- Relational Model:

dm(<u>senderID</u>, <u>receiverID</u>, <u>messageID</u>)

FK: senderID references User
receiverID references User
messageID references Message

2.2.3 Writes

- Relational Model:

writes(<u>userID</u>, <u>messageID</u>, <u>roomID</u>)

FK: userID references User
messageID references Message
roomID references Chatroom

2.2.4 Follows

- Relational Model:

follows(<u>userID</u>, <u>followerID</u>)

FK: userID references User
followerID references User

2.2.5 Admin

- Relational Model:

admin(userID, roomID)

FK: userID references User
roomID references Chatroom

2.2.6 Contain Bet

- Relational Model:

contain_bet(slipID, betID)

FK: slipID references bet_slip
betID references Bet

2.2.7 Publish About

- Relational Model:

publish_about(newsID, matchID)

FK: newsID references News
matchID references Match

2.2.8 Played On

- Relational Model:

played_on(betID, matchID)

FK: betID references Bet
matchID references Match

2.2.9 Create Betslip

- Relational Model:

create_betslip(userID, slipID)

FK: userID references User

slipID references Betslip

2.2.10 Arranges

- Relational Model:

arranges(slipID, bet_type)

FK: slipID references bet_slip

bet_type references betting_type

2.2.11 Written On

- Relational Model:

written_on(commentID, commentableID)

FK: commentID references Comment

commentableID references Commentable Entity

2.2.12 Reply

- Relational Model:

reply(commentID, commentedCommentID)

FK: commentID references Comment

commentedCommentID references Comment

2.2.13 Consist of

- Relational Model:

consist_of(leagueID, teamID)

FK: leagueID references League
teamID references Team

2.2.14 Played By

- Relational Model:

played_by(matchID, home_teamID, away_teamID)

FK: matchID references Match
home_teamID references Team
away_teamID references Team

2.2.15 Like/Dislike

- Relational Model:

like_dislike(userID, likedEntityID, like_dislike, like_count, dislike_count)

FK: userID references User
likedEntityID references Likeable Entity

3.Implementation Details

In this project we had used PHP and MySQL on PHPMyAdmin which is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. We had stored our data on our local computers and by using XAMPP we connected our database to our application. We designed our GUI by using bootstrap in there we designed our web pages. We had added 4 leagues to our datasets 2 for football and 2 for basketball. In each leagues there are 10 teams and we declared 3 matches and 2 bet slips including these matches. Currently there are 2 users in our system and they commented an liked to existing matches and to each others profiles.

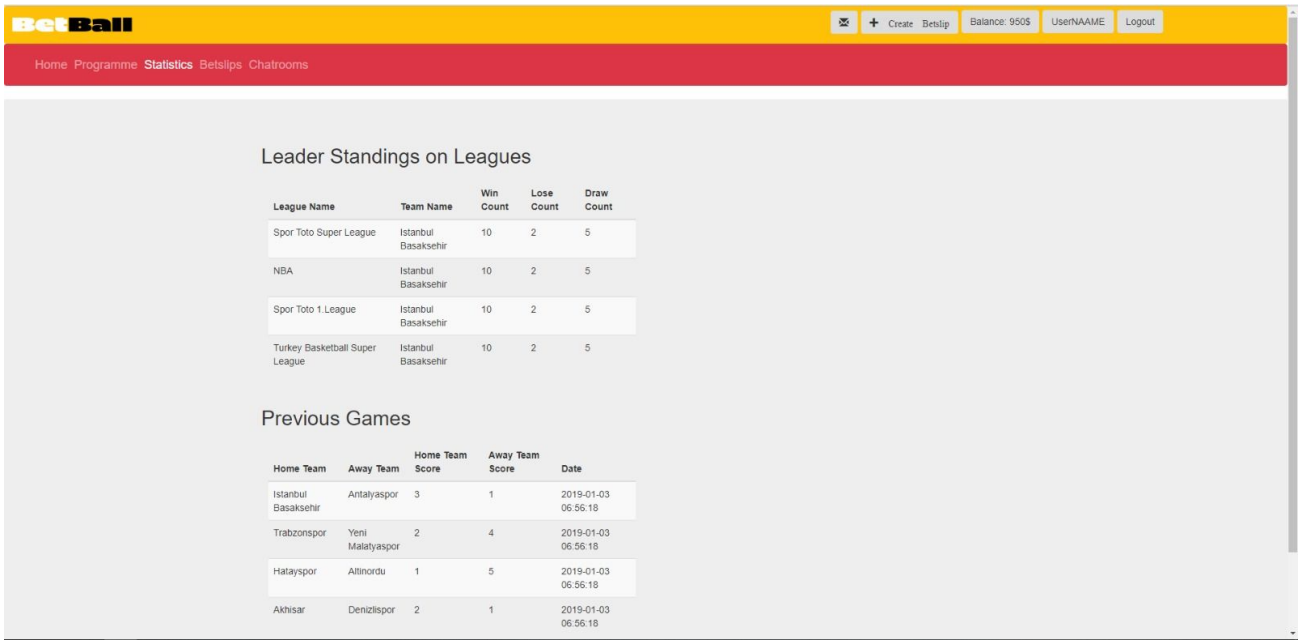
In order to make an user interface, we used HTML and CSS to form the website and Bootstrap to arrange the menus and pages. As we developed the website, we initially created headers and the footer to make a template for all the pages. After that, we arranged the navigation between the pages. When we completed the pages, we started to put the bodies of the pages one by one.

Biggest challenge so far was connecting front end and back end. Receiving matches form the database, displaying them in the betslip creation page and making a bet slip was the hardest. Other than that, we have managed to do the connection more easily. In order to learn Bootstrap we used w3schools[1] website to check and improve our user interface.

4.Sample Outputs

In betslip page users can see the standing and upcoming matches and their odds. By using those information they can create their betslips. Those betslip information will include total ratio and total winning.

The statistics page contains information about standing and previous games.



5. Advanced Database Components

5.1 Views

5.1.1 Teams with biggest average predictions

This view orders the average predictions of teams in an ascending order and prints them. By looking at this, user can see which teams are powerful because lower average prediction means more powerful team.

```
CREATE VIEW teams_Average_predictions AS
    SELECT * FROM team ORDER BY average_prediction ASC LIMIT 5;
```

Teams with biggest average predictions

ID	Name	Win Count	Lose Count	Average Prediction
11	Gencler Birligi	13	2	1.02
10	Akhisar	4	8	1.02
1	Istanbul Basaksehir	10	2	1.05
21	Fenerbahce beko	11	1	1.05
31	Bucks	26	10	1.05

5.1.2 Leader Standings on Leagues

It shows the leaders of each league with their win and lose counts.

```
CREATE VIEW league_leaders AS
    SELECT L.name x, T.name y, T.win_count z, T.lose_count t, T.draw_count m
    FROM league L, team T, consist_of C
    WHERE C.teamID=T.teamID AND C.leagueID = L.leagueID
    GROUP BY L.leagueID
    ORDER BY T.win_count DESC
```

Leader Standings on Leagues

League Name	Team Name	Win Count	Lose Count	Draw Count
Spor Toto Super League	Istanbul Basaksehir	10	2	5
NBA	Istanbul Basaksehir	10	2	5
Spor Toto 1.League	Istanbul Basaksehir	10	2	5
Turkey Basketball Super League	Istanbul Basaksehir	10	2	5

5.1.3 Most Liked and Commented Betslips

These views stays in the statistics page and shows the most liked and most commented bet slips.

```
CREATE VIEW most_liked_slips AS
        SELECT a.slipID x, a.result y, a.betting_amount z, a.created_date t,
b.like_count
        FROM bet_slip a, likeable_entity b
        WHERE a.likeableID=b.likeableID
        ORDER BY like_count DESC

CREATE VIEW most_commented_slips AS
        SELECT c.userID x, c.username y, b.text z, b.date t
        FROM written_on a, comment b, user c, bet_slip d
        WHERE d.slipID='$slip' AND d.commentableID=a.commentableID
                AND     b.commentID=a.commentID AND c.userID=b.userID
```

Most Liked Betslips

Slip ID	Result	Betting Amount	Date	Like Count
2	0	10	2019-01-03 04:19:24	1
1	1	100	2019-01-03 04:19:06	0

Most Commented Betslips

User ID	Username	Text	Date
2	Leona	I think this is a very risky odd	2019-01-03 03:46:23

5.2 Reports

5.2.1 User Monthly Report

Users will be presented with a report of their total losses/wins and how many bets did they lose with how many matches so that they can see how close they were to winning and keep playing.

$userID is kept in the session when the user logs in and keeps the userID.

```
CREATE VIEW monthly_user_report AS
    SELECT result
    FROM (Betslip JOIN create_betslip as T)
    WHERE T.userID=$userID and
    DATEDIFF(created_date, CURRENT_TIMESTAMP()) < 30
    GROUP BY slipID;
```

5.2.2 New Comments Report

This report will be given to admin so that he/she can examine the comments and delete the foul ones.

```
CREATE VIEW new_comments_report AS
    SELECT text
    FROM Comment
    WHERE DATEDIFF(date, CURRENT_TIMESTAMP()) < 30;
```

5.3 Triggers

- When a Betslip is created related User account should be updated.
- When a Match has ended related Team and League tables should be updated. Moreover betslip's that predicted this match's result false should be finalized and marked as lose and user's win_ratio should be updated.
- When a Comment or Betslip got liked/disliked their likes/dislikes should be updated.
- When a User starts following or getting followed

5.4 Constraints

- Betting amount cannot exceed the balance of the user and can't be smaller than the slip's minimum bet amount.
- Users cannot join to a Chatroom if it is full.
- Count of bets can't be smaller than the slip's minimum bet count.

6. References

[1] (n.d.). Retrieved December 30, 2018, from https://www.w3schools.com/bootstrap/default.asp