Premier League Project

Progress Report

CS3200: Database Design

Blackboard Group:
DunbarEliasFang

Team Members:
Ben Dunbar
Filmon Elias
Jike Fang

Professor & TA information:
Dr. Kathleen Durant
Akash Singh
Priyank Kumar

Table of Contents

Introduction

Our project is to build a website that provides users with information about the Premier League: a club based English soccer association. The website will provide users with a personalized page to view particular clubs and/or players of their choice. A single external view will provide this functionality whereas the conceptual level and internal level will hold the static information about the league. The users will be able to make actions and modify data to interact with the website. In particular, the Users entity (defined below) can create, read, update and delete data via relations to the Follows, Ratings and Comments entites.

The project is of interest to the group because we enjoy soccer as a sport and are followers of the Premier League. However, as soccer fans there is a lack to the amount of personalized connection with the League, which this project aims to resolve.

In terms of sharing information, users of the site will have a public and private profile. In this way they can share information such as which teams they follow, which players they have rated, and which comments they have posted on games. However, their personal information will not be shown on the public profile to keep them safe. A reach goal of the project can be to also add friendships to the users so that it is easier to share their information.

User Interaction

Users will first have to log on to use the site. After logging on the users will be able to update basic information about themselves if they choose to. While logged on the users can view information about clubs, players and games and other users. These pages will be linked in such a way that from a club page, you can visit the pages of the players of that club and the games that club has played. From a player's page you can navigate to the club for which the player plays and games in which the player has had events occur, such as scoring a goal. From a game page, you can navigate to either of the clubs involved in the game. From a game page you can also navigate to any players page if that player was involved in an event in the game, such as scoring a goal. At all of these pages the user will be able to perform an action, such as: following a club, rating a player, and commenting on a game. Furthermore, these three types of pages will be reachable via navigation bars on the page so that a user who isn't following any clubs yet can still find content. Please note, a user's profile will be considered their home page.

Additionally, users can navigate to other users public profiles via comments they have made. The public profiles will not display any unwanted personal information. The data shown could be fetched from a view which does not have sensitive information. The figure below gives an overview of the flow of actions available to users. Note that additional interactions could be added in if the team has extra time and implements friendships and messages between users. There are many possible flows of actions for a user, so the user interaction diagram has been

made to show a connected network of page views with arrows for possible navigations and actions (in Bold).

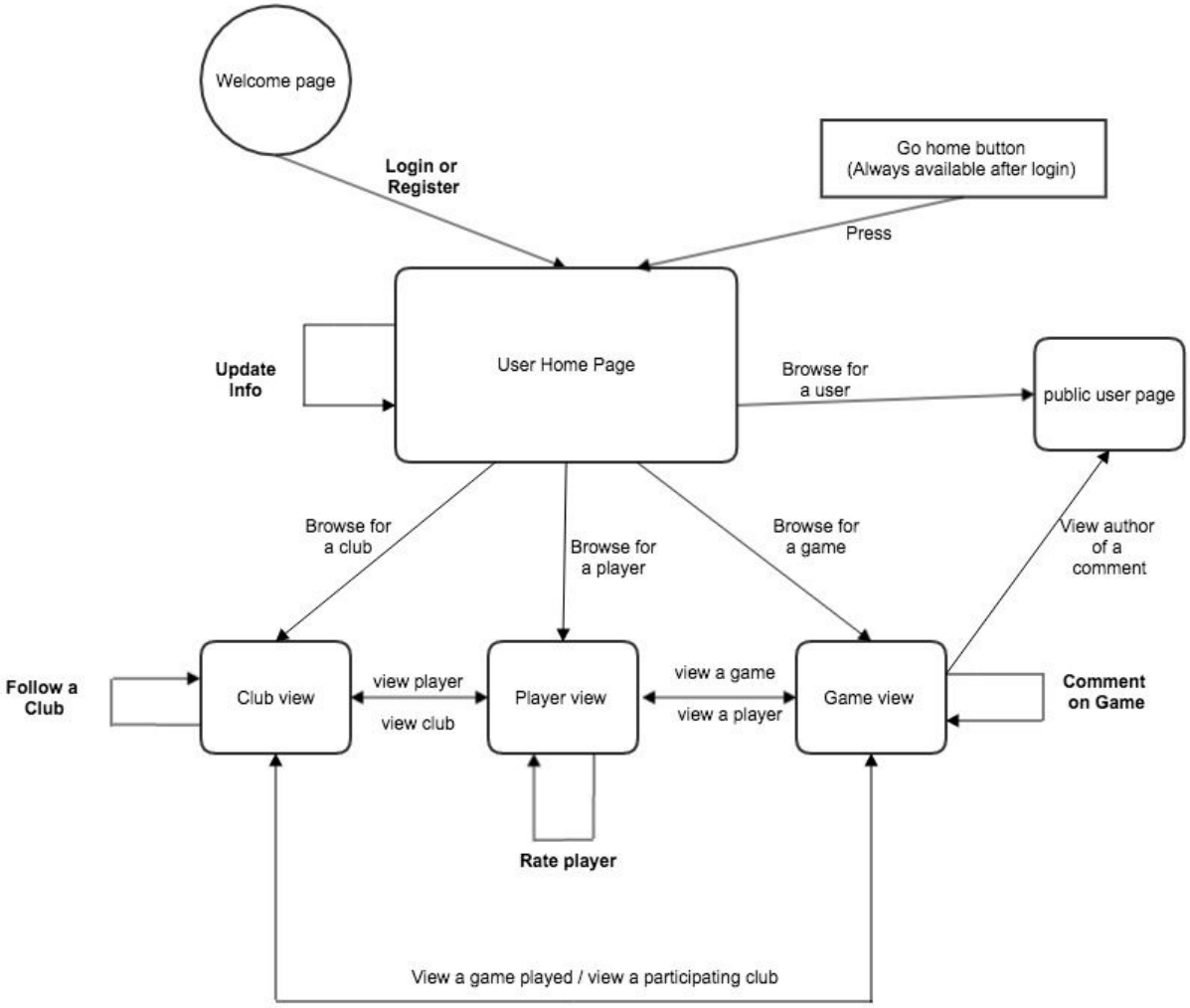

Figure 1: User Interactions

Technical Specifications

The framework of the Premier League Project will be split up into three different parts. There will be a website to act as a user interface, a server to provide the website with the ability to interact with the database, and a database to store all of the information required by the project. This will allow for a demonstration of the team's ability to work with a system while using a database for storage. The project will use a Node.js and MySQL framework.

The plan for the frontend is to use JavaScript, HTML, and CSS in order to provide a user friendly interface for an enjoyable experience. Some of this technology is new to the team, but as a three member team it should still be manageable within the scope of the project. The front end will access data via a server. The proposed library to make the front end organized and professional is Angular.js version 1.6.

The server will act as the middle tier of the project and will provide the frontend with the ability to execute operations. The team will use Node.js with Express.js version 4.15 to create this connection. The team will use the mysql node package module as described in the Express guide to integrating a MySQL database.

Finally, the team will use MySQL for the DBMS of the project. This will align with the topics in class and allow for a robust storage system. The database will have multiple entities, such as: Clubs, Players, Player_Contracts, Games, Events, Event_Types, Users, Follows, Ratings and Comments.

The team believes this framework will be reliable and documented online enough for the team to learn and work with. In the event of migrating from a locally hosted website to a public website, the team can use Amazon web services (Elastic Beanstalk).

Data Acquisition

In order to load data about the premier league into the project itself, the team has done some research and found a free API at Football Data.org which has endpoints that can provide most of the desired data. The data used for this project will be all of the data available for the Premier League 2016/2017 season. After looking into the how the Football Data organization names and offers data, the primary endpoints to be used are:

- Clubs in Premier League: http://api.football-data.org/v1/competitions/426/teams
- Games in Premier League: http://api.football-data.org/v1/competitions/426/fixtures
- Players in Premier League: http://api.football-data.org/v1/teams/322/players
    - Used once for each of 20 teams in league

Notes Regarding football-data.org:
The Premier League is referred to as a "competition" with id 426
The clubs are referred to as "teams"
The games are referred to as "fixtures"

The data responses from these endpoints are in a JSON format and can be manipulated to provide the specific fields required. The data for events such as goals and red cards, etc. does exist online, but was not easily provided from a free API. Additionally, the event table has the possibility of containing thousands of rows, but this is not necessary to develop nor showcase the project. Only a few entries need to exist in order for the project to be operational. In order to keep the event table from lacking data, the project team will add events from their favorite games. The site Scoreboard.com will be used to look up the events in the games. An example page is located at: http://www.scoreboard.com/game/manchester-city-liverpool-2016-2017/8WvrM48M/#game-summary|game-statistics;0|lineups;1.

Entity Relationship Diagram

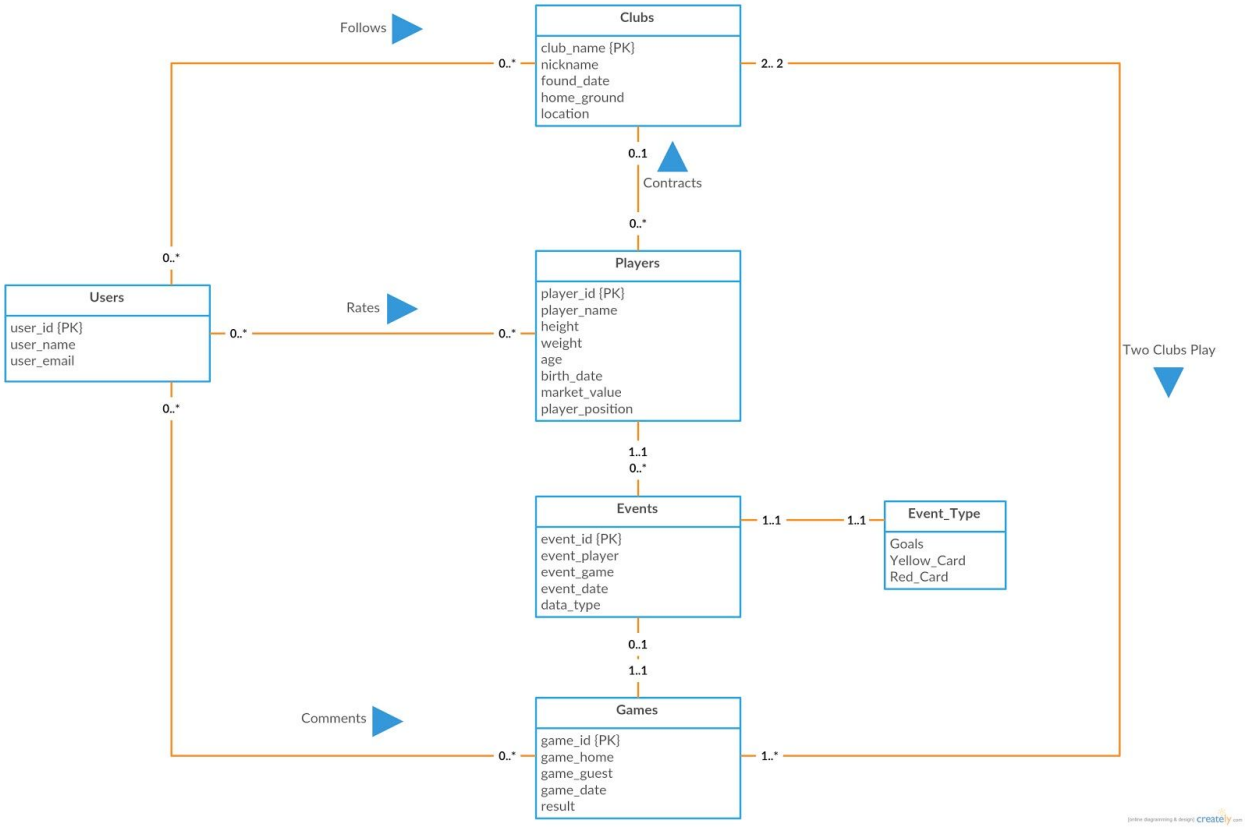

Figure 2: Premier League ERD

The ERD diagram given in Figure 2 will allow for a holistic representation of the Premier League while following good database design practices. The model should allow for avoidance of data traps like the fan trap and chasm trap and it should help reduce duplication of data compared to other approaches. This model allows for user interactions with the data via following and rating different clubs or players. Additionally, users will be able to interact with the site by commenting on games to provide their thoughts.

EER Diagram

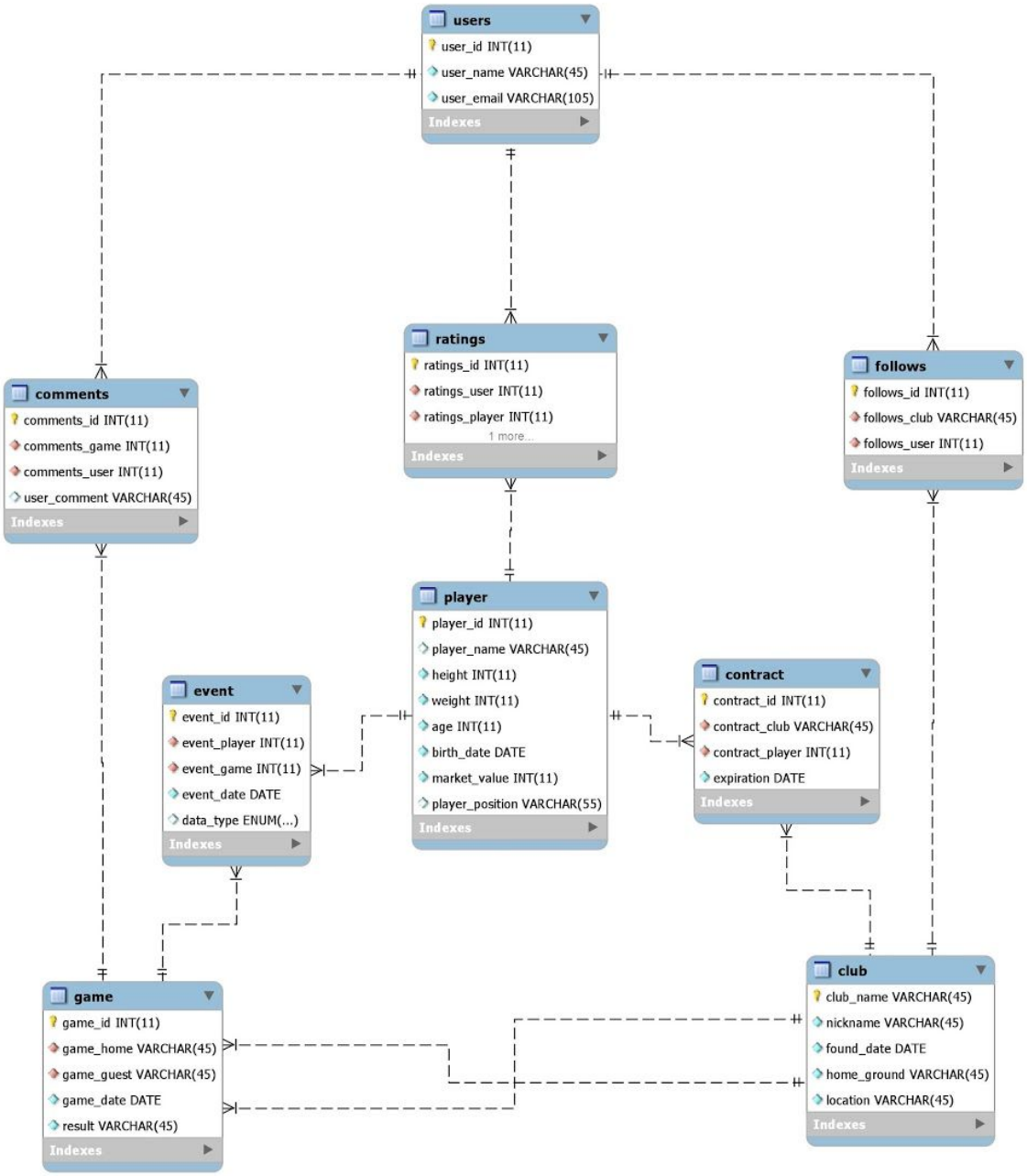

Figure 3: Premier League EER

The team created the initial SQL script for the database and generated all of the tables. After creating the tables, the team has developed figure 3 to model the database. This model shows how comments, ratings, and follows have also been chosen to be stored in tables.

Conclusion

Our group contains 3 people since the database we are going to design is large and we decided to design a website as well. There are also some new technologies we need to learn first like JavaScript for the website. In terms of changes from the proposal we have decided to focus on JavaScript for the majority of the project. Instead of creating a Java backend, we have taken some advice from another classmate on Piazza and will use Express.js with our Node.js project. For these reasons we think our group size is appropriate for the proposed Premier League project. We look forward to continuing our work on the database and website.

References

Express with MySQL:
https://expressjs.com/en/guide/database-integration.html#mysql

AWS with Node.js:
https://aws.amazon.com/developers/getting-started/nodejs/

Angular.js Guide:
https://docs.angularjs.org/guide

API Site for Data Acquisition:
http://api.football-data.org/index

Site for event data collection:
http://www.scoreboard.com/game/hull-city-leicester-2016-2017/K2bWNGpt/#game-summary|game-statistics;0|lineups;1