CS 353
DATABASE SYSTEMS
Spring 2018

ÇiftKale
A Football Database System

PROJECT PROPOSAL DOCUMENT

Prepared By:

Hakan Türkmenoğlu
Orkun Alpar
Selim Fırat Yılmaz
Alper Şahıstan

Table of Contents

Introduction

This document proposes a database application called ÇiftKale which is a football database system. Contents of this report includes the problem description, features of the system, some design limitations and requirements. Database system will be presented using both an E/R diagram and a verbal description.

Problem Description

People are in greater need of looking up and tracking through their favorite players, clubs and leagues. They are also highly interested in transfers news and happenings in the world of football. Instead of browsing through various web pages, we are suggesting an integrated system.

Project Definition and Features

ÇiftKale is a football management database system which will be used by football fans and managers. It will include information about players, agents, clubs, leagues and coaches. It will also include news about transfers, match results along with statistics(such as possession, shoots, shoots on target, corners, passing, fouls and cards). It'll also have league standings.

It allows it's users to create profiles and track statistics about their teams and players of interest. It's functionalities will also include setting values for players by managers, make transfer offers for players and accept/reject offers for them. Clubs will have a transfer budget and an annual wage for each player.

Sportsman is the generalization of Player and Coach. Sportsman profiles will include a picture of the sportsman, name, age, nationality, current team, previous teams, national appearances annual wage and if possible, social media accounts. Main distinction between coaches and players is different statistics. Apart from those mentioned, Player profiles might also include stats from FIFA and/or PES.

Clubs will have a logo, name, country, squad and a transfer budget.

Leagues will have standings and statistics about teams & matches.

Matches will have teams, time, date, statistics, and a venue.

Users of the system will have an email address, name and a hashed password. "User" denotes a fan, a standard account. "Users" will have 2 specializations, namely, "director" and "administrator".

Extended Features

We have some other features in mind but we are not sure whether we'll be able to implement them or not. Hence, we wanted to write those ideas down as a separate section. Time allows, these features will also be included in the system:

- Comment section under posts, where fans can discuss.
- Virtual betting
- An artificial intelligence implementation that would simulate a league using the data of players and previous matches. For the training of AI, we'll use a sequence-to-sequence model such as recursive neural network. Players of the simulation will be able to put offers and make transfers, which would affect the simulation dynamically.

Approach

This project requires a strong grasp of database management as it deals with tons of information mentioned above. Therefore, in order to have it work efficiently; relations, entities, attributes and other aspects of the database should be implemented meticulously. ÇiftKale is a web application and it will constantly make connections with the database, it will create, read, update and delete data.

Requirements Specification

Functional

Sign Up – Sign In

Visitors should be able to sign up to the system and maintain that profile, stored in our database. They should then be able to sign in to their profiles. Standard user profile will hold:

- User name (unique)

- Email address

- Hashed password

- Phone number (optional)

- Teams followed (set by user)

Director means the director of a club. It is another type of an account. Directors can make transfer offers and accept/reject offers directed to the players of their clubs.

Administrator (admin) has the highest priority among users. An admin can add/update/remove entities, except the case that it interferes with the internal affairs of clubs.

Transfer offers

As mentioned above, directors of clubs can make offers. They can also accept/reject them. These offers made will be displayed through the news feed. A director can not offer more than the budget of his club.

News Feed

News feed will serve as a medium for having users see the offers for players. Offers will have five parameters:

- Offering club

- Player of interest

- Offered club

- Amount offered

- Status of the offer (pending, declined, accepted)

Administrator Panel

As mentioned, admins will have the right to edit the database. Therefore, a separate panel will be provided.
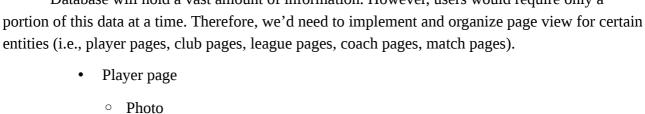
Home Page

System will have a landing page. Contents of this page will include:

- News feed

- Upcoming matches

- Recent matches

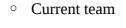- League standings

- Search bar (Players, Teams, Leagues, Coaches)

Data Display via Pages

Database will hold a vast amount of information. However, users would require only a portion of this data at a time. Therefore, we'd need to implement and organize page view for certain entities (i.e., player pages, club pages, league pages, coach pages, match pages).

- Player page
  - Photo
  - Name
  - Age
  - Position
  - Nationality
  - Current team
  - Previous teams
  - Statistics
- Coach page
  - Photo
  - Name
  - Age
  - Nationality
  - Current team
  - Previous teams
- Club page
  - Logo
  - Name
  - Short name
  - League
  - Squad
  - Director
  - Previous matches
  - Upcoming matches

- League page
  - Logo
  - Name
  - Teams
  - Standings
  - Previous matches
  - Upcoming matches
- Match page
  - Home team
  - Away team
  - Score
  - Venue
  - Time & Date
  - Statistics

Limitations

- User name should be alphanumeric and unique.
- Users will not be able to delete any content.
- Administrators can not interfere with the internal affairs of clubs.
- Directors, however can edit player teams according to transfer offers.

Non-Functional

Constraints

- MySQL 2nd Gen 5.7 will be used as database engine.

- Python will be used for backend development.

- HTML, Javascript, and CSS will be used for frontend development.

Reliability

System should be reliable. It should be consistent and let it's users trust in it.

Extendability

We aim to extend this system in the future. Therefore, system needs to be extensible.

Accessibility

System should be user friendly. There shouldn't be anything that's highly complex for regular users.

Security

System should be secure. It should not let unauthorized users access certain features.

Performance

Queries should be responded in minimum amount of time. System should be highly fast.

E/R Model