Contents

Managerial Football Simulation

Abstract

Managerial football simulation is a software program that allows a passionate footballer to take the seat of a coach to a football club of choice, perform managerial duties, take the team to the field and see how it goes. Managerial duties range from tactics, match day line ups, club finances and transfers.

Based on the player attributes, the team's tactics, the opposition players and tactics, a full football match is simulated and all the important events and statistics are recorded.

One can create a full season with various leagues and cup competitions, and can even invite friends to take part in the season. Each season has an auto generated game calendar that mimics the real life football calendar, with fixtures set according to these dates. But in real life time, there are three game calendar weeks in a day. So the manager gets to coach 3 games or 4 in a day. After a season is over, the creator can renew the season and also edit some data, add or remove friends ahead of the new season.

One can also create a custom tournament and invite friends to take part, taking up different teams chosen for the tournament and then get a fixture list generated and start playing.

One can also take part in transfer activities, making requests and getting accepted or rejected feedback from the other party, and complete the signings and include them in their teams before the closure of the transfer period.

This game also intends to include teams and data from the local leagues and others from around the region, in an intention to promote the local leagues and football growth. At the moment, no existing games online or on consoles include local leagues and local football data about Kenya, or any other country in the region in entirety.

One can create a season for just KSH 40/- and renew a season for KSH 20/- . A custom tournament would cost 45/- . One can request for more transfer money at extra costs depending on how much transfer money the manager requests. Plus managers pay just KSH 5/- per game managed. Everyone has a financial account indicating how much the user has at their disposal. We intend to have a pay bill number that one can use to send money to the online account.

This document contains the design specifications of the system data and content creation and management, rather than the simulation algorithms.

SYSTEM MODULAR DESIGN

The system is to be divided in specific modules:

1. User data and processes
2. Game calendar
3. Tactics module
4. Competition and fixtures
5. Transfer module
6. Season module
7. Financial module

User data and processes

The user needs to provide some basic information when registering: names, email, preferred username and password. Then a user's auto-generated id is then used to track a user's activity. On registration, a confirmation link is sent to the user's email address, before any login can be allowed.

On login, only the preferred username and password are required.

Game calendar

This mimics the typical football calendar, from August to May, with match day fixtures typically in the weekends and once a while during midweek. We would have a running daemon at the server to keep updating a timestamp on the database every ***n*** hours with a timestamp mimicking a weekday or a match day.

Tactics Module

This module deals with the team's formations, lineups, game philosophy and players' position on the field of play. Each team has a particular formation, joined on a ***team_id*** column on tables: **teams** and **team_formation**. Each **team_formation** has assigned a particular player assigned to a particular position in the field indicated by **init_x** and **init_y** positions, meaning initial x and initial y position coordinates respectively on table: **player_positions**, joined on field - **id** of table **team_formation** on column **team_formation_id**. There are various predefined formations and are in the table: **tactic_formations** joined on table **teams** on field: **formation** to field **id** on the **tactic_formations** table. The engineer has to design algorithms to auto calculate the players initial x and y coordinates based on the formation chosen.

Competitions and Fixtures

Teams are organized in their respective leagues with fixtures and results are encapsulated. The engineer has to design algorithms to auto generate fixtures of the leagues and tournaments. A team indicates membership to a certain competition through an entry in the **competition_teams** table adding its id in the **team_id** column. Each competition is associated with a fixture through the **competition_id** field in the **fixture_list** table. Each fixture has stats, like ball possession, shots on goal, etc. through the **fixture_id** column on the **fixture_stats** table.

Transfer Module

A manager can make transfer of a player from one team to the manager's team. A transfer entry is made into the **transfers** table and more details respective to the fields in the **transfers** table. The **game_calendar** field in the **transfers** table has to be a value between **from_stamp** and **to_stamp** in the **transfer_window** table. A daemon would be required to keep updating the transfer window table to activate transfer windows by updating the **from_stamp** and **to_stamp** columns. A manager can request can request for extra transfer cash, but will be charged accordingly, say 15/- per 10 million pounds. This is where integration with the financial module is required, with a maximum request amount of 100 million pounds.

Season Module

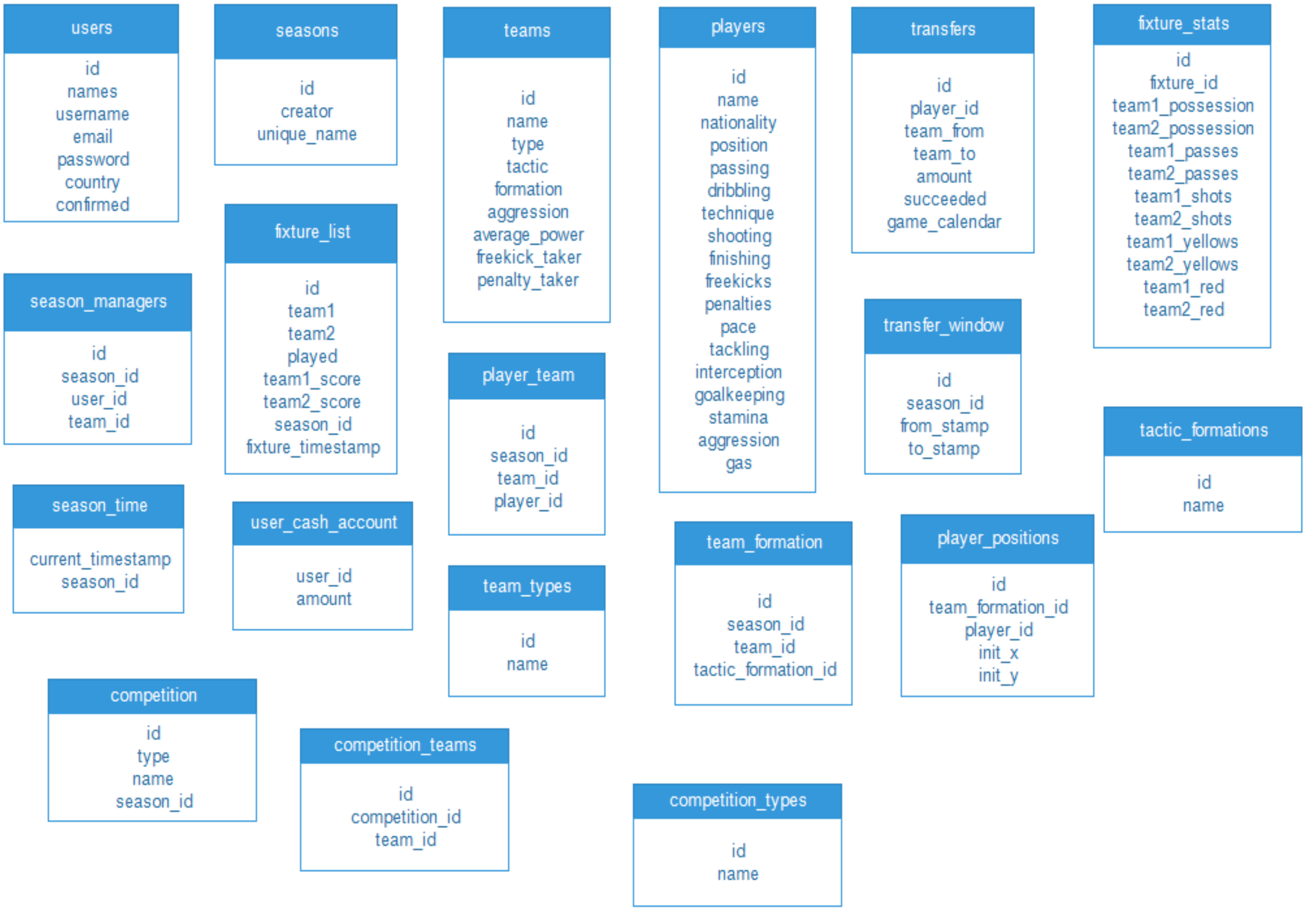
A season namespaces all entities: competitions, team tactics, transfers and fixtures. A season identifies who is playing and it's the primary entity that namespaces a group of friends playing together. Most tables have a **season_id** column for identification purposes, because seasons share teams, players and competitions but other entities like fixtures, tactics are internal to a season, to avoid collision. Competitions are global but sometimes need a **season_id** column if it's a custom created competition.

Financial Module

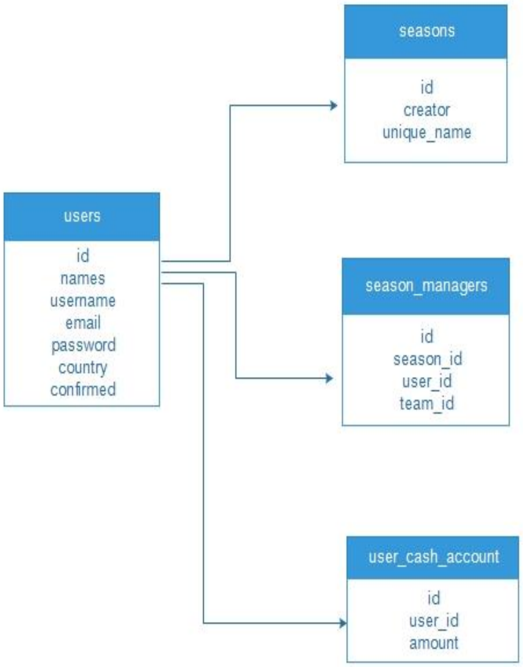
This provides specifications on how the entire system can be monetized. This provides the functionality of our pay bill number, where the money is forwarded to our pay bill account and the amount is reflected on the user's account, through the **user_cash_account**.

ENTITY RELATIONSHIP DIAGRAMS

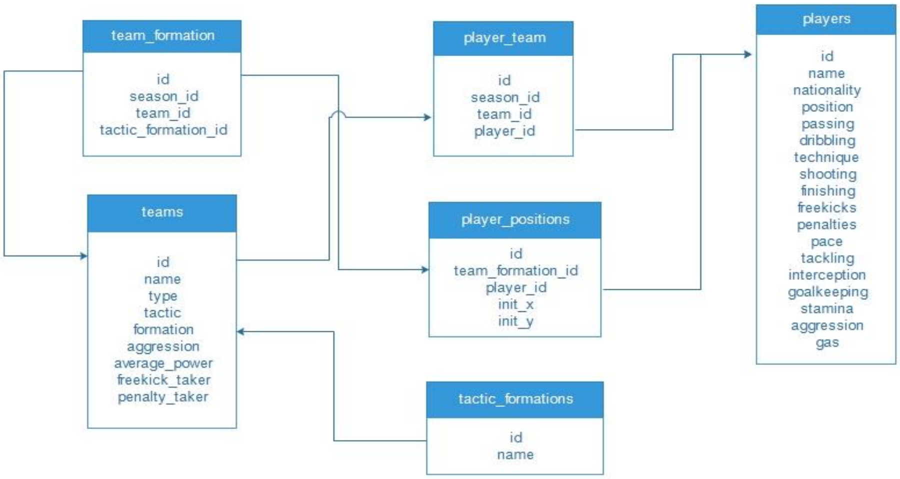
1.0 All Entities



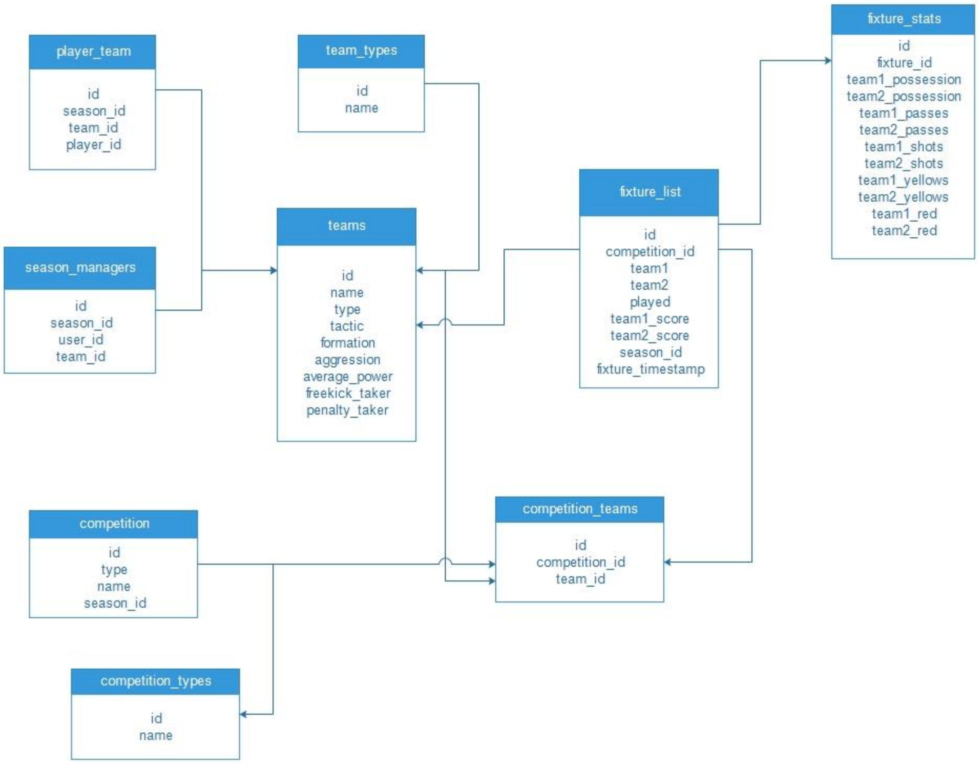
2.0 Users – Seasons relationship



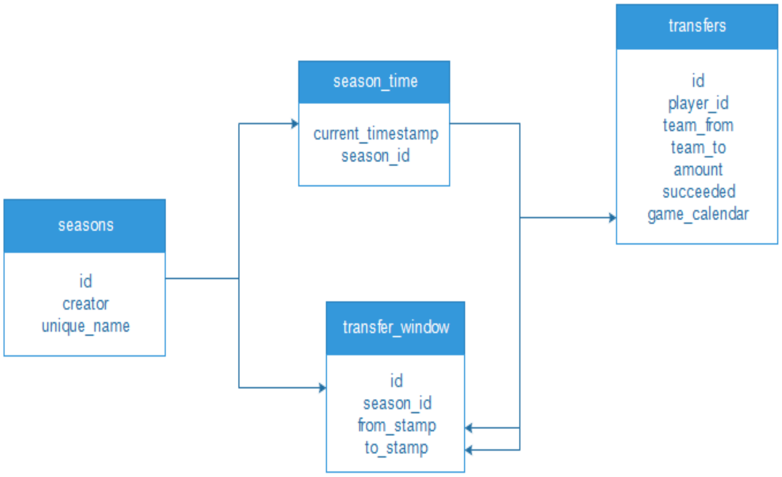
3.0 Team – Player – Tactic relationship



4.0 Team – Competition – Fixture relationship



5.0 Time – Transfer Relationship


transfers
id
player_id
team_from
team_to
amount
succeeded
game_calendar
season_time
current_timestamp
season_id
seasons
id
creator
unique_name
transfer_window
id
season_id
from_stamp
to_stamp