**User Requirements Document  
FOOTBALL MANAGER**

Eddy Ekofo - X00119959  
Graham Lalor - X00122026  
Philip Cheape - X00123757  
Juris Cicelimovs - X00116645

Contents

1. **Introduction 2**
   1. **Survey 2**
2. **System Narative 5**
3. **Use Case Diagram 6**
4. **Use Case Descriptions 7**
   1. **Login 7**
   2. **Register 8**
   3. **Manage Team 10**
   4. **Play Match 11**
   5. **Edit Team 12**
5. **Class Diagram 14**
6. **Pseudo Code 15**

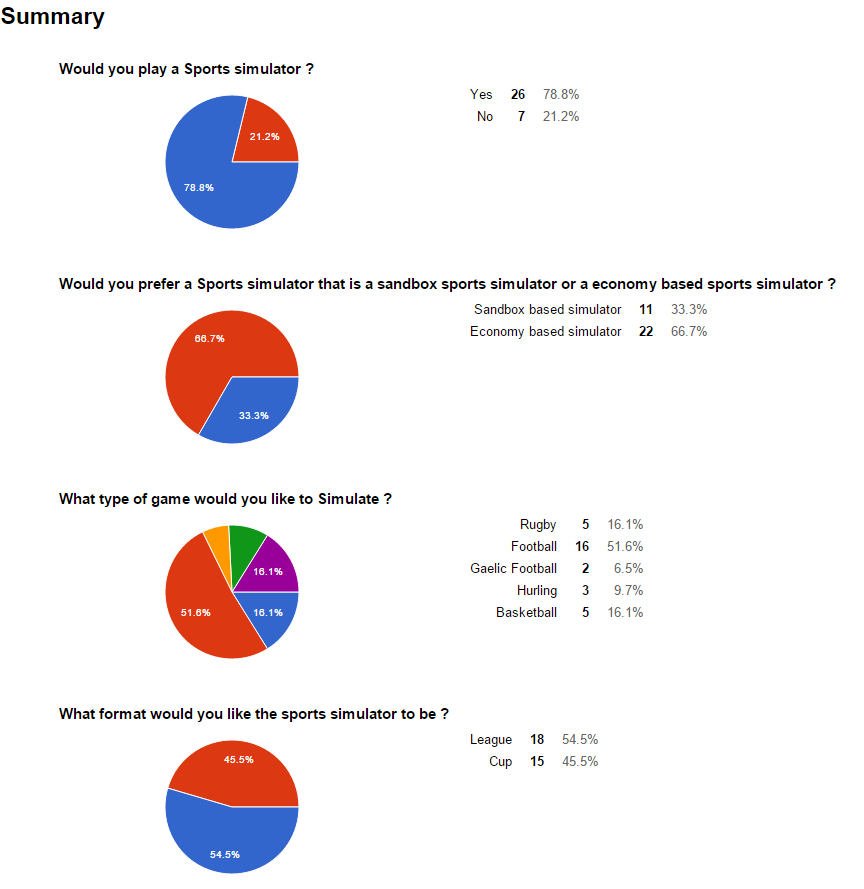
**Introduction**

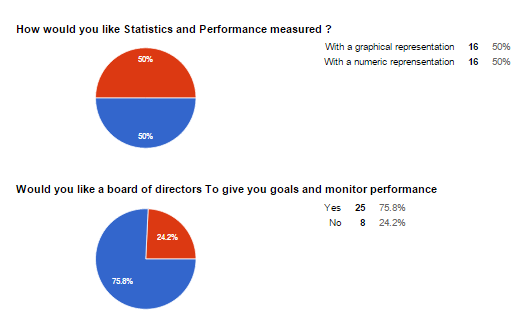
This is the user requirements documentation on the football management game. It gives the initial outline on how the base system will be develop in order for the user to be able to play a base game. Once the base system is developed then the goal would be to develop an economy and to give the user goals in the form of a board of director’s feedback.

**Survey**

1. Would you play a Sports simulator?
2. Would you prefer a Sports simulator that is a sandbox sports simulator or an economy based sports simulator?
3. Would you prefer a Sports simulator that is a sandbox sports simulator or an economy based sports simulator?
4. What type of game would you like to Simulate?
5. What format would you like the sports simulator to be?
6. How would you like Statistics and Performance measured?
7. Would you like a board of directors to give you goals and monitor performance?
8. Would you like the ability to simulate the training of your players?

The results of the Survey were as follows:





From these questions the following requirements were determined.

**Must have**

* Login
* Register - create account
* Manage Team
  + Player Injuries
  + Training
  + Player Statistics
  + Editor - edit team Name, Players name.
* League
* Website
* Match Results

**Nice to have**

* Economy
* Stadium
* Transfers
* Board of directors

**System Narrative**

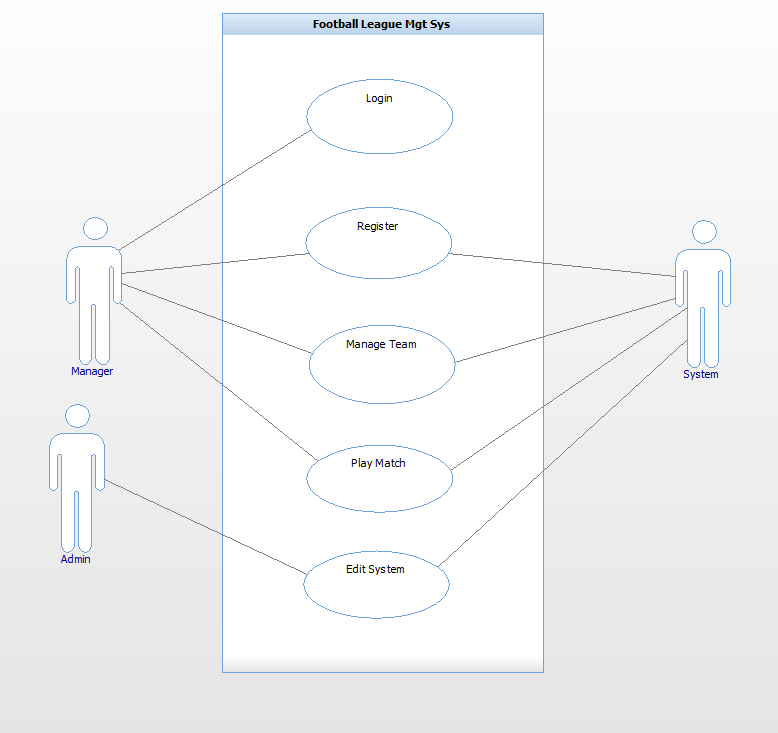
After reviewing the results of our survey, we know that there is a potential market for a Sports simulation 78.1% of the twenty-five people to participate in our survey said they would be keen to use a system like ours. In our survey we asked which sport our potential users would prefer our system to focus on, 51.6% of people said they would like the system to focus on football. Users would like the ability to create unique accounts and be able to sign in to their account to use our management system users will only be able to manage one team.

In the Editor the Users will be able to change the player names and team name at any stage throughout the league.

With this system users will be able to create team line ups, add players or remove them if they find they are not as good as other potential players along with improving players with training and monitoring injuries .This is the Team Management functions.

The System will automatically calculate the match results and update the fixtures with the result as each of the matches are played.

We have broken down our user requirements into two categories this is because we found that the majority of our potential users would like to include the ability to simulate games and a league's online between other users (insert %). We have a must have category, this category contains all the features our users would like to have in our system and that we find are necessary for our system to be unique and successful.

Use Case Diagram

**Use Case Description**

**Login**

|  |  |
| --- | --- |
| Use Case: | Login |
| Actor(s): | Manager, System |
| Goal: | To successfully log into the system within the constraints provided. |
| Overview: | This allows the manager to log into their account. In order to have a team each manager must first be registered to access the system the manager must be logged in at first. |
| Pre- Condition: | 1. Must be a registered manager 2. Must provide the correct username & password |
| Post- Condition: | 1. Successfully logs into the system |
| Successful Scenario: | 1. System prompts username & password fields 2. Manager accurately enters username & password 3. System validates the provided details 4. Manager successfully logs into the account |
| Alternative Scenario(s) | 1. Manager provides incorrect password & username System notifies manager. |

**Register**

|  |  |
| --- | --- |
| Use Case: | Register |
| Actor(s): | Manager, System |
| Goal: | To successfully register and provide the required details. |
| Overview: | This allows the manager to register and own an account on the system. If the manager does not have an account they can register to own one. |
| Pre- Condition: | 1. Must provide the required details |
| Post- Condition: | 1. Successfully registers into the system. |
| Successful Scenario: | 1. System requests details 2. Manager provides username 3. Manager provides password 4. Manager provides Team Name 5. Manager provides his Name 6. System registers the manager. |
| Alternative Scenario(s): | 1. Manager provides unavailable username. System prompts for new manager name.   4. Team name is already in use. System prompts for new Team Name. |

**Manage Team**

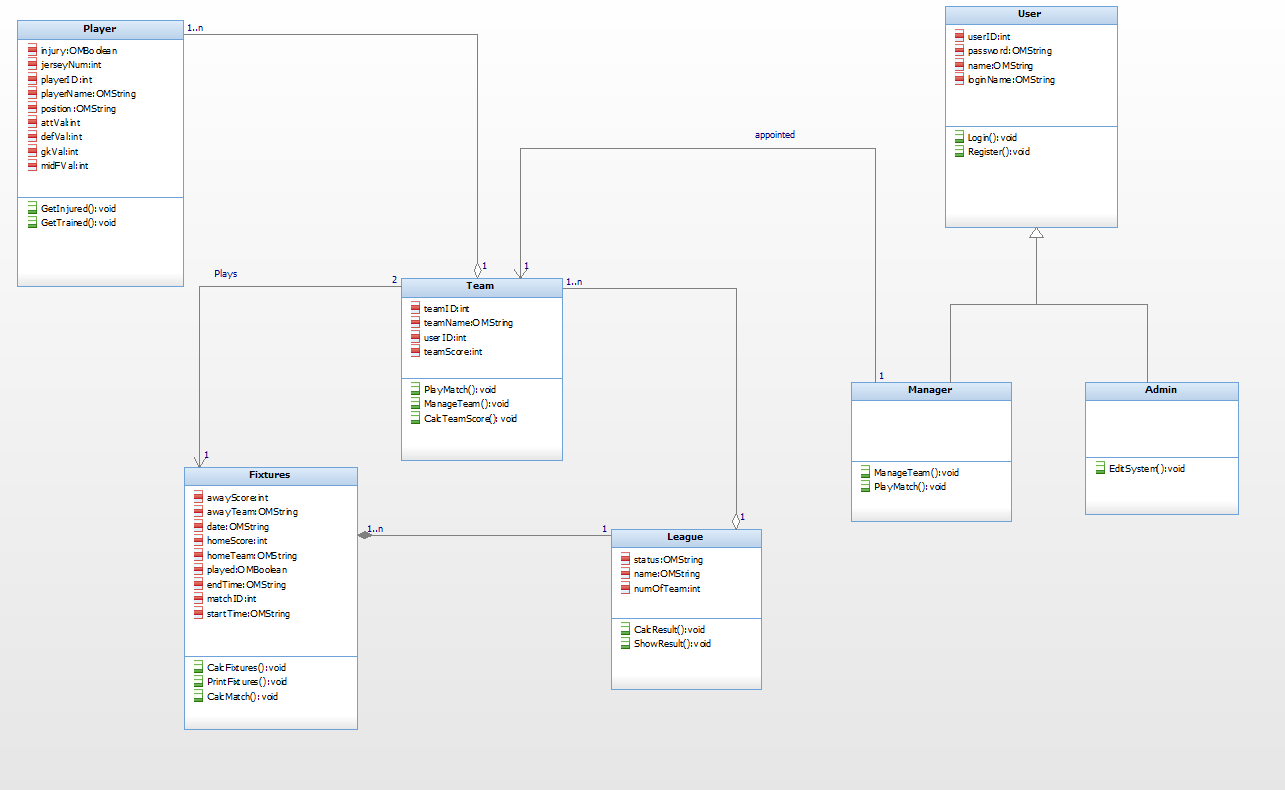
|  |  |
| --- | --- |
| Use Case: | Manage Team |
| Actor(s): | Manager, System |
| Goal: | To manage their own team. |
| Overview: | It's allow Manager create team, edit team players, edit their ability. Training - to increase a player’s performance. |
| Pre- Condition: | System updates the team So manager can view current player statistics |
| Post- Condition: | None |
| Successful Scenario: | Step 1. Manager Clicks on field he wishes to edit. (training, positions , etc)  Step 2. Manager clicks on Accept at bottom of field.  Step 3. System checks that all entries are valid  Step 4. System updates database. |
| Alternative Scenario(s): | Step 3. Any invalid fields System notifies manager to change. |

**Play Match**

|  |  |
| --- | --- |
| Use Case: | Play Match |
| Actor(s): | Manager, System |
| Goal: | To successfully play match and get results as the outcome. |
| Overview: | This allows the manager to play a match which the system simulates results, the outcome is either a win, draw or lost. |
| Pre-  Condition: | 1. Each team must have the required number to complete a squad of 11 before the match can be played.  2. Must be played within the boundaries provided by the system (date, time, home/away etc.) |
| Post-  Condition: | 1. Successfully plays the match, the system provides the outcome of the game. |
| Successful  Scenario: | 1. System provides match day fixtures.  2. Manager enters the exact number of their squad  3. System stimulates the match and provides the results |
| Alternative  Scenario(s): | 2. Manager submits an insufficient number of squad; the system prompts the manager to submit the sufficient required amount before the match proceeds. |

**Edit Team.**

|  |  |
| --- | --- |
| Use Case: | Edit Team |
| Actor(s): | Admin |
| Goal: | To successfully Edit team, change name, train team. |
| Overview: | This allows the admin to edit team name, player stats and details. |
| Pre-  Condition: | N/A |
| Post-  Condition: | N/A |
| Successful  Scenario: | 1. Administrator logs in  2. Chooses team name  3. Changes play/team details |
| Alternative  Scenario(s): | 2. Chooses user details  3. Changes user details |

**Class Diagram** 

**Pseudo Code CalculateMatch()**

**Inputs:**

teamaScore Int

teambScore Int

**Outputs**

WinnerScore int

LoserScore int

**Constants**

Name: Type

**Other**

Name: Type

randTeamscoreA (1-100) int

randTeamscoreB (1-100) int

randDrawScore(0-3) int

loserScore(0-10) int

totalTeamScoreA

totalTeamScoreB

Begin

generate randTeamScoreA(1-100)

generate randTeamScoreB(1-100)

(teamaScore + randTeamScoreA)/10 = totalTeamScoreA

(teamBScore + randTeamScoreB)/10 = totalTeamScoreB

begin if

totalTeamScoreA = totalTeamScoreB

then randDrawScore = winnerGoalScore

winnerGoalScore = loserGoalScore

end if

begin if

\\calculate winners score teamAscore > teambBScore

TeamteamAScore - teamBScore = WinnerScore

\\calculate losers score goalKeeperScore= get winning teams goalkeeper attribute add random 1-10

attackScore = get loser teams attak and random 1-10

begin if

goalKeeperScore - attackScore > 0

rounded up then LoserScore = (goalKepperScore-attackScore)/2

end if

End