Second Year Project

FOOTBALL MANAGER

2016



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

# Table of Contents

[Table of Contents 2](#_Toc449027880)

[User Requirements Document 4](#_Toc449027881)

[Introduction 5](#_Toc449027882)

[Survey 6](#_Toc449027883)

[Results 7](#_Toc449027884)

[Requirements 9](#_Toc449027885)

[System Narrative 10](#_Toc449027886)

[Use Case Diagram 11](#_Toc449027887)

[Use Case Description 12](#_Toc449027888)

[Login 12](#_Toc449027889)

[Register 14](#_Toc449027890)

[Manage Team 16](#_Toc449027891)

[Play Match 18](#_Toc449027892)

[Edit Team 20](#_Toc449027893)

[Class Diagram 22](#_Toc449027894)

[Pseudo Code 23](#_Toc449027895)

[Design Specification 24](#_Toc449027896)

[Website Story Board 25](#_Toc449027897)

[Website Screenshots 27](#_Toc449027898)

[Home Page 27](#_Toc449027899)

[Login Page 28](#_Toc449027900)

[Register Page 29](#_Toc449027901)

[Fixtures Page 30](#_Toc449027902)

[Squad Page 31](#_Toc449027903)

[League Table 32](#_Toc449027904)

[ERD Diagram 33](#_Toc449027905)

[Table/Record Layout 34](#_Toc449027906)

[Basic System 34](#_Toc449027907)

[User 34](#_Toc449027908)

[Player 34](#_Toc449027909)

[Team 35](#_Toc449027910)

[League 35](#_Toc449027911)

[Fixtures 36](#_Toc449027912)

[TeamFixtures 36](#_Toc449027913)

[Advanced System 37](#_Toc449027914)

[User 37](#_Toc449027915)

[Player 37](#_Toc449027916)

[Team 38](#_Toc449027917)

[League 38](#_Toc449027918)

[Fixtures 39](#_Toc449027919)

[TeamFixtures 40](#_Toc449027920)

[Stadium 41](#_Toc449027921)

[Bank 41](#_Toc449027922)

[Test Case Matrix 42](#_Toc449027923)

[Pseudo Code 45](#_Toc449027924)

[calculateMatch() 45](#_Toc449027925)

[addTrainVal() 47](#_Toc449027926)

[calculateResults() 49](#_Toc449027927)

[getTrained() 51](#_Toc449027928)

[manageTeam() 53](#_Toc449027929)

[User Manual 55](#_Toc449027930)

# User Requirements Document

## Introduction

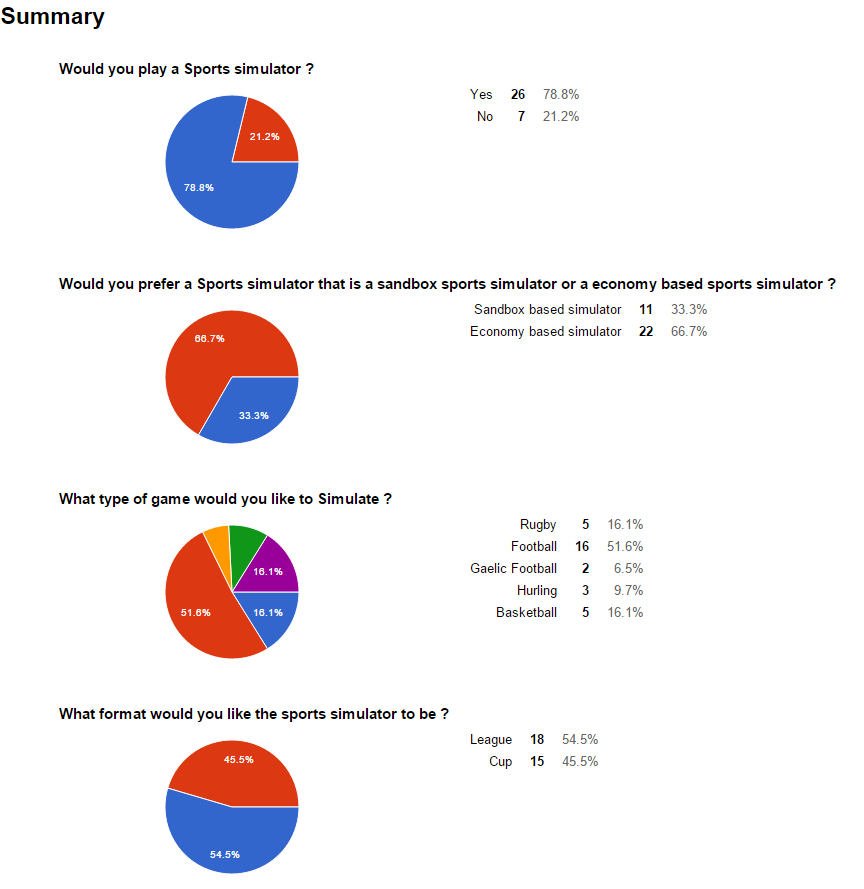
This is the user requirements documentation on the football management game. It gives the initial outline on how the base system will be develop in order for the user to be able to play a base game. Once the base system is developed then the goal would be to develop an economy and to give the user goals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

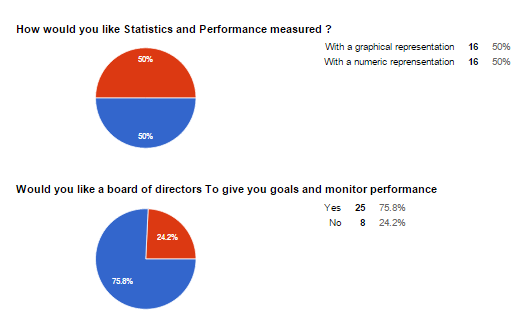
## Survey

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

### Results

The results of the Survey were as follows:





### Requirements

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.

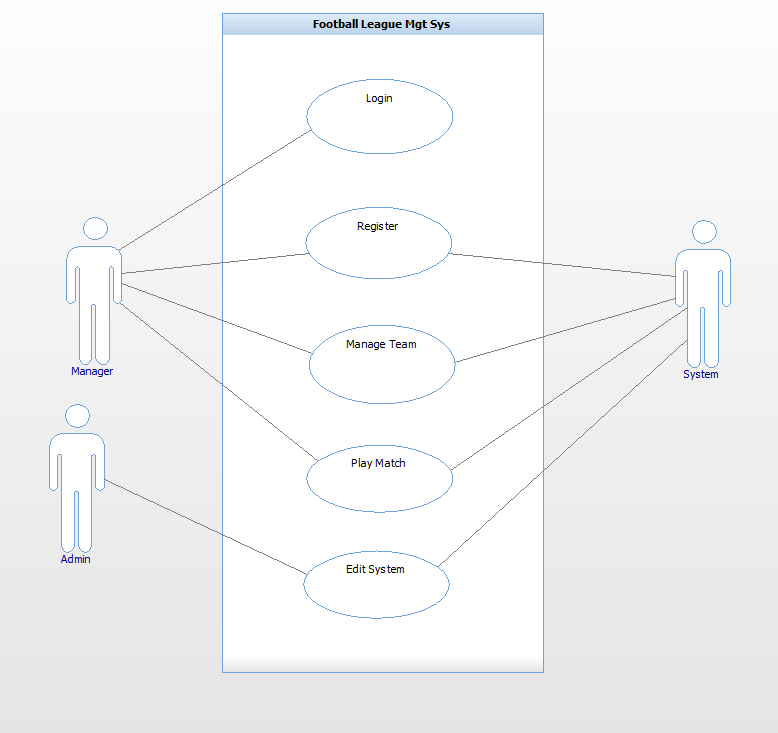
In the Editor the Users will be able to change the player names and team name at any stage throughout the league.

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: | Must be a registered manager  Must provide the correct username & password |
| Post- Condition: | Successfully logs into the system |
| Successful Scenario: | System prompts username & password fields  Manager accurately enters username & password  System validates the provided details  Manager successfully logs into the account |
| Alternative Scenario(s) | 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: | Must provide the required details |
| Post- Condition: | Successfully registers into the system. |
| Successful Scenario: | System requests details  Manager provides username  Manager provides password  Manager provides Team Name  Manager provides his Name  System registers the manager. |
| Alternative Scenario(s): | 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. |

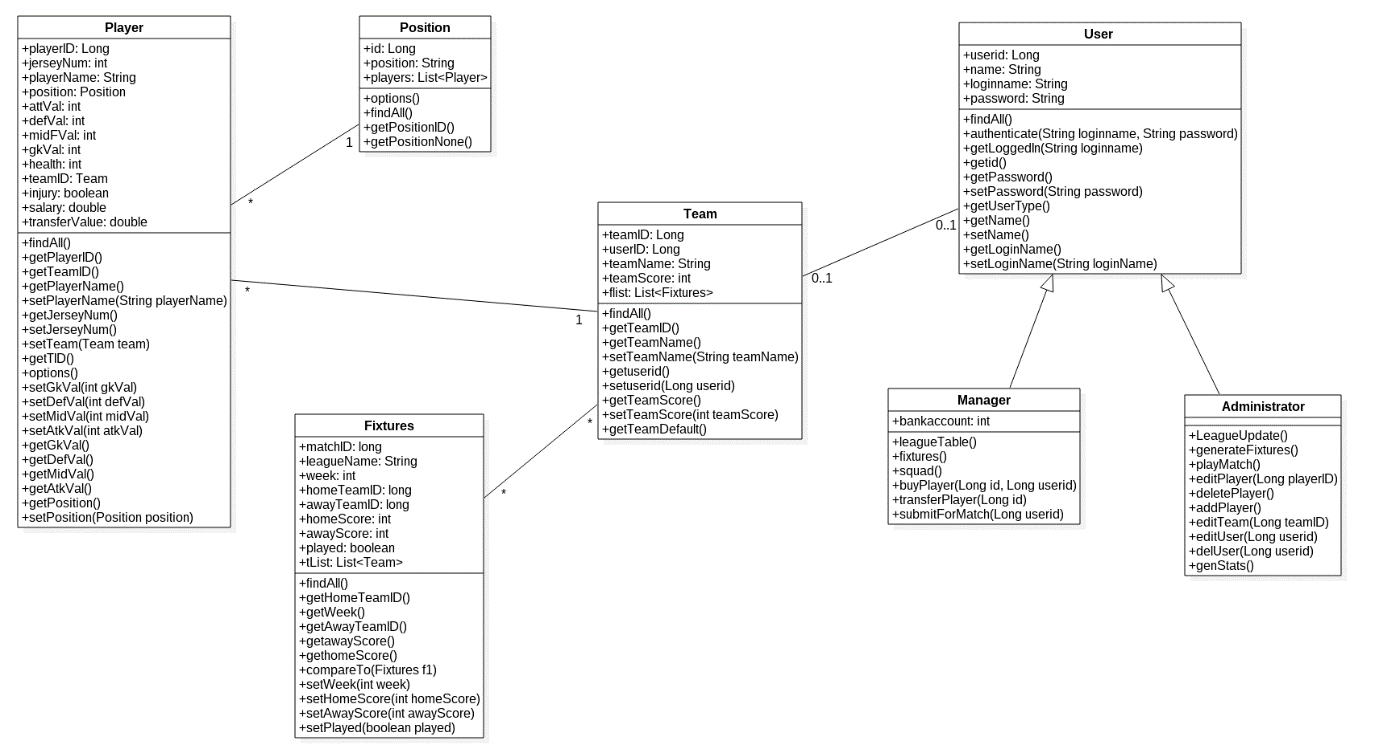
## Play Match

|  |  |
| --- | --- |
| 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

**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

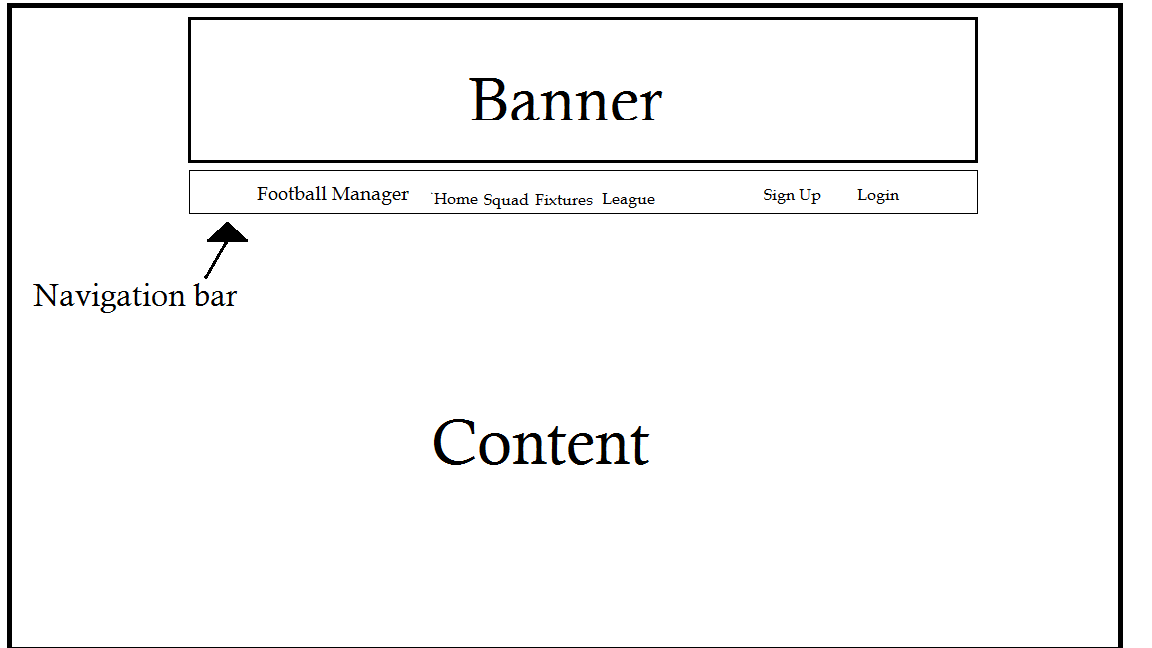
# Design Specification

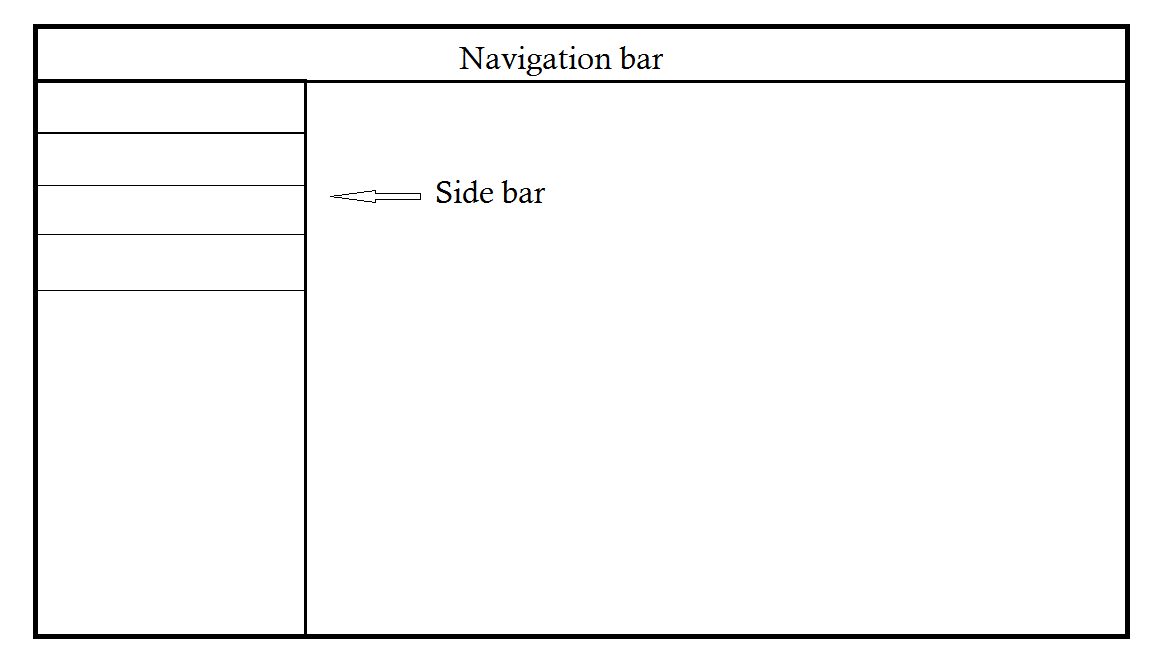
NOTES:

We decided to include the ERD Diagram and table/record layout for both our basic system and advance system but under different heading. This is because we plan on implementing this functionality if we have enough time.

# Website Story Board

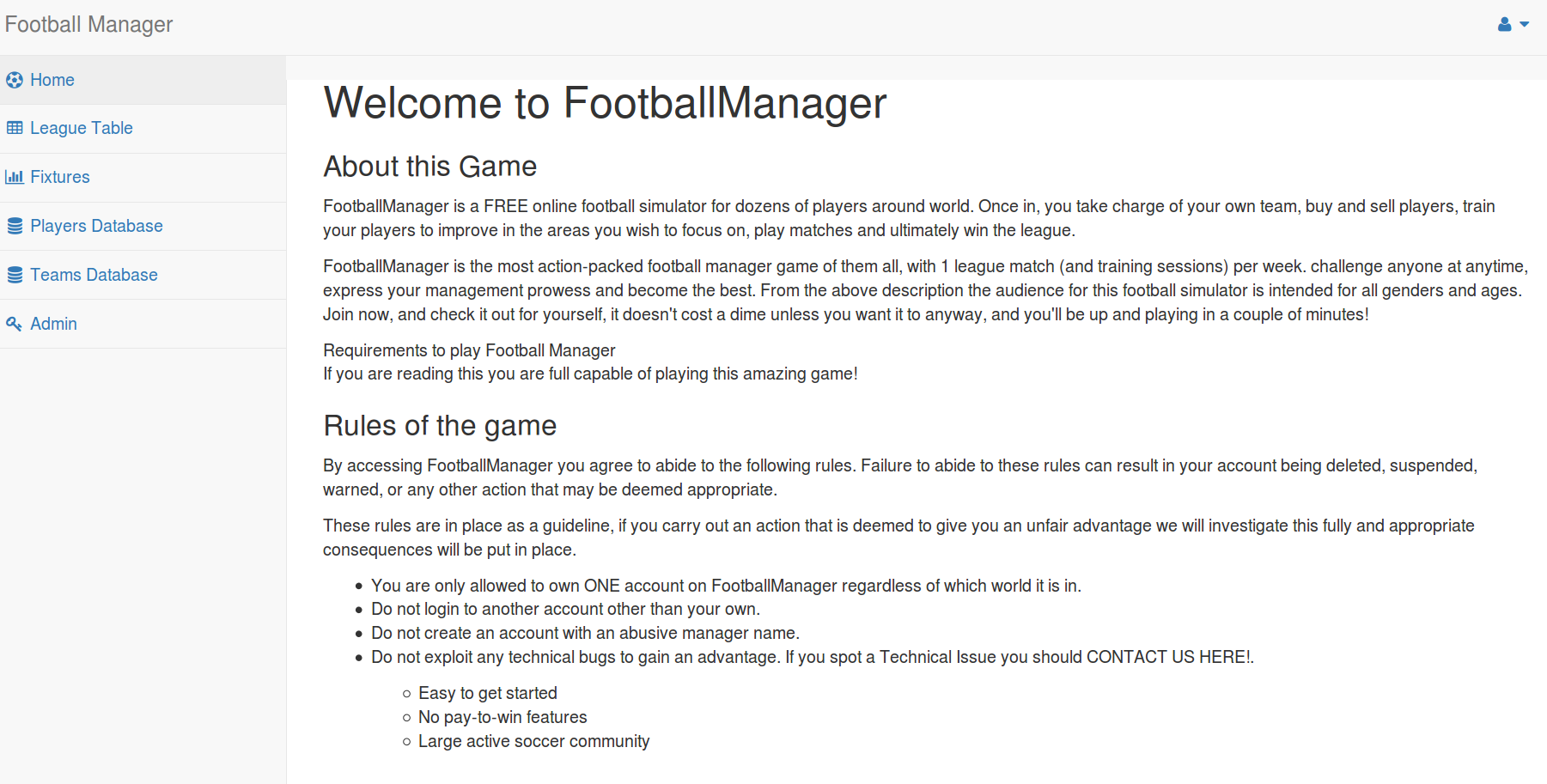
These are the patterns in which we will base the design of our website. The reason why this was chose is due to its simplicity and clarity to present the contents which will be displayed in our website.



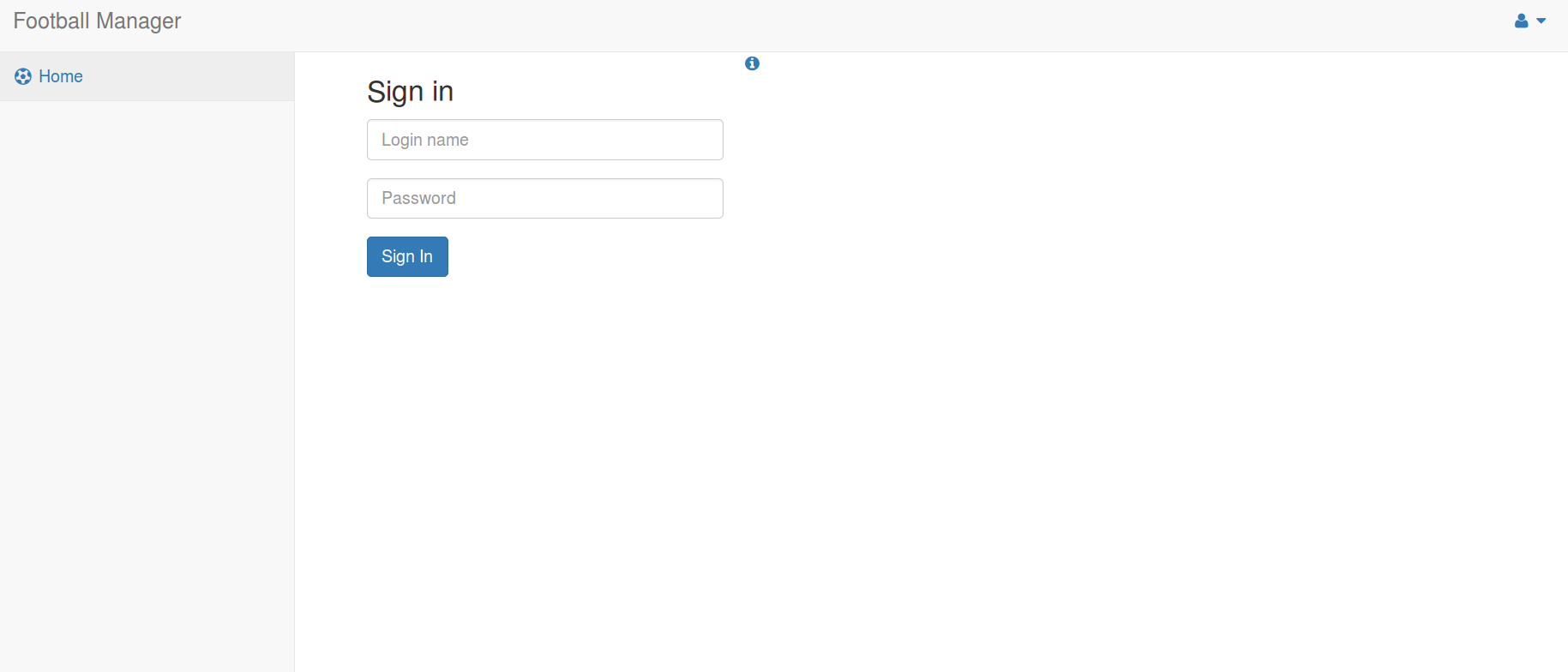


# Website Screenshots

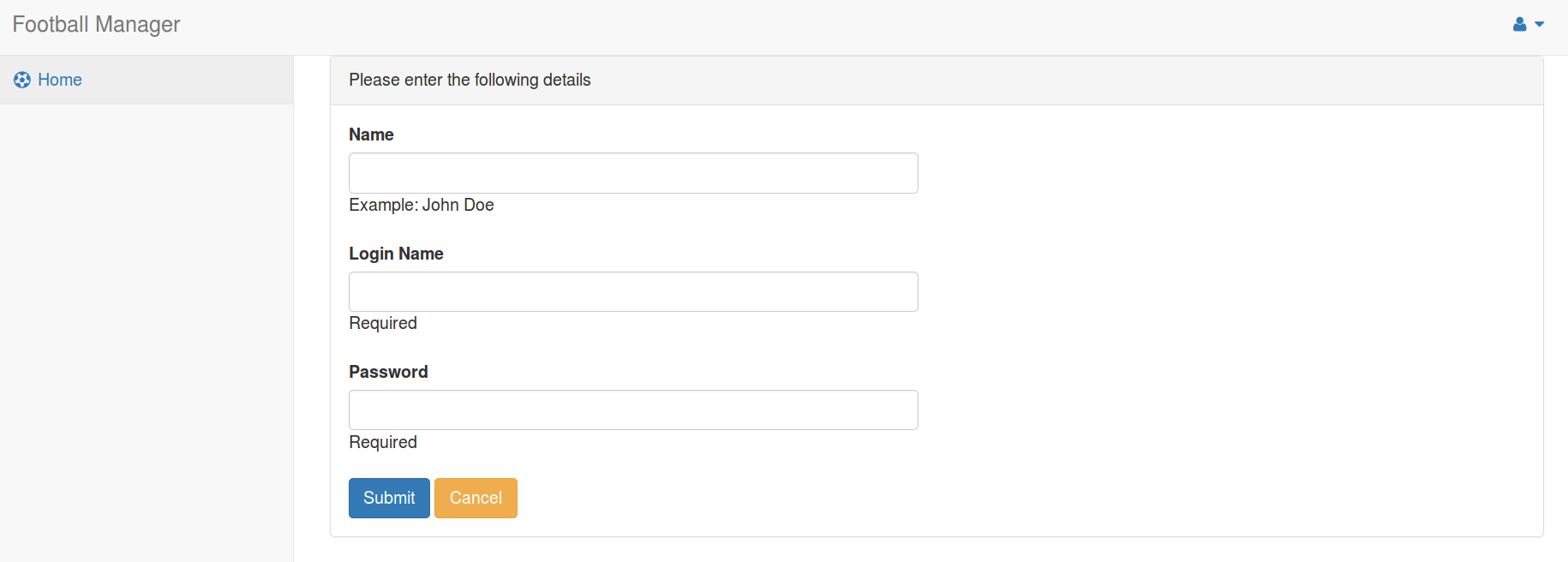
## Home Page



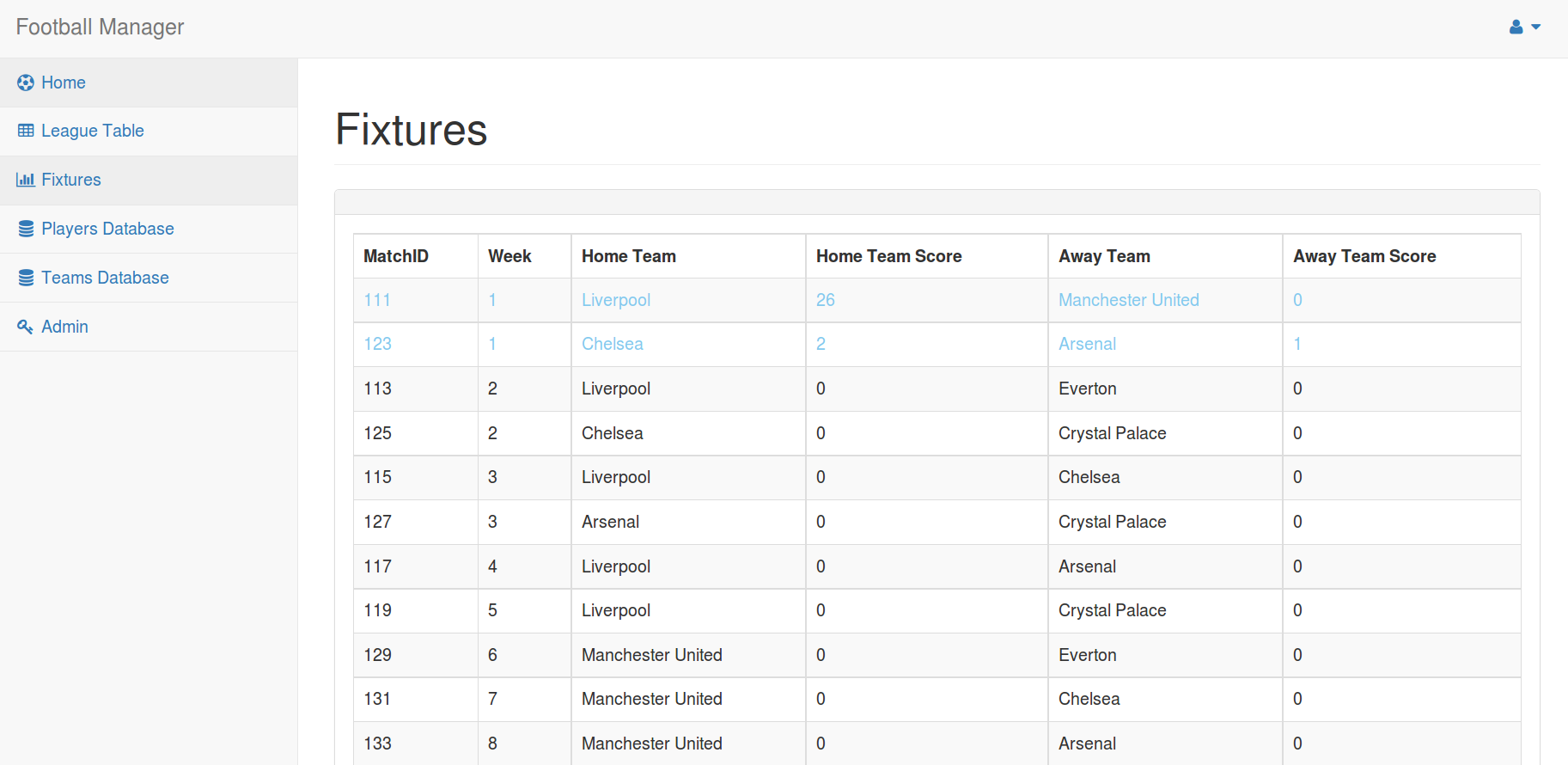
## Login Page



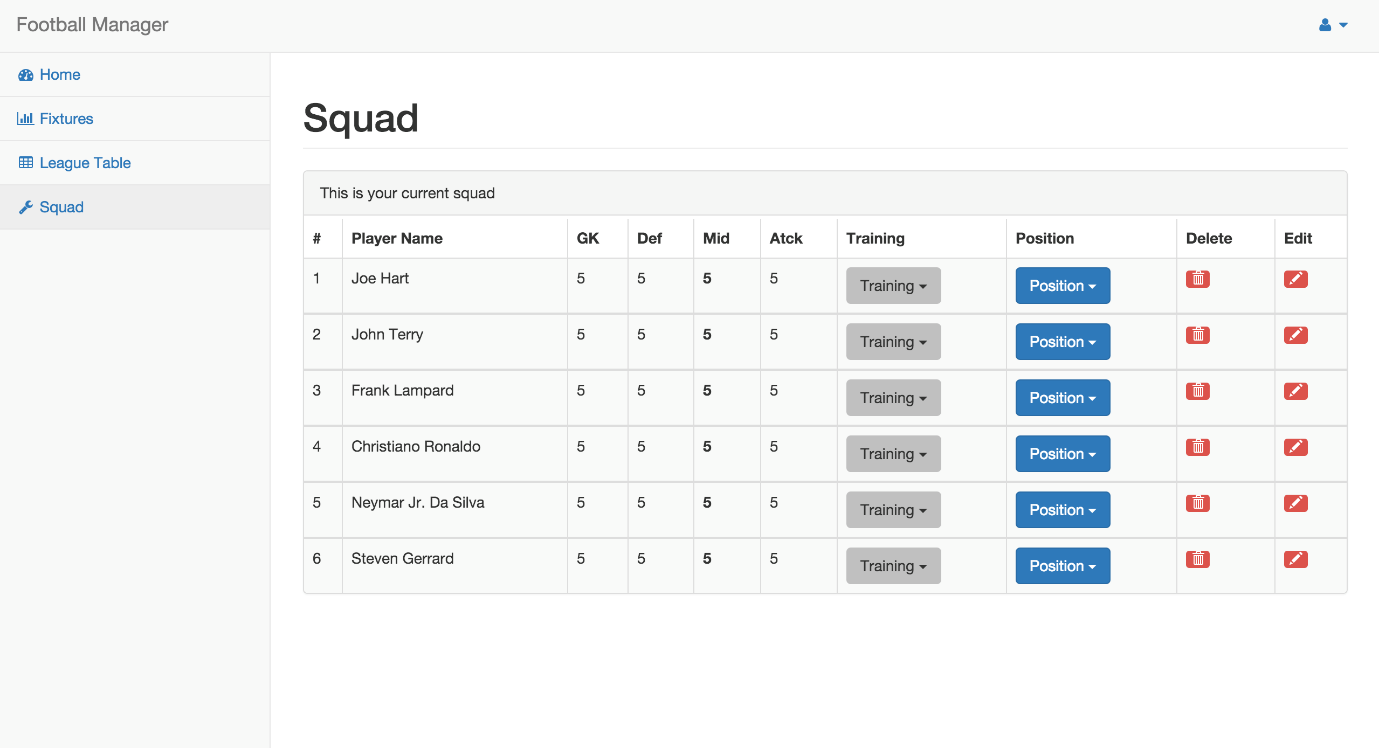
## Register Page



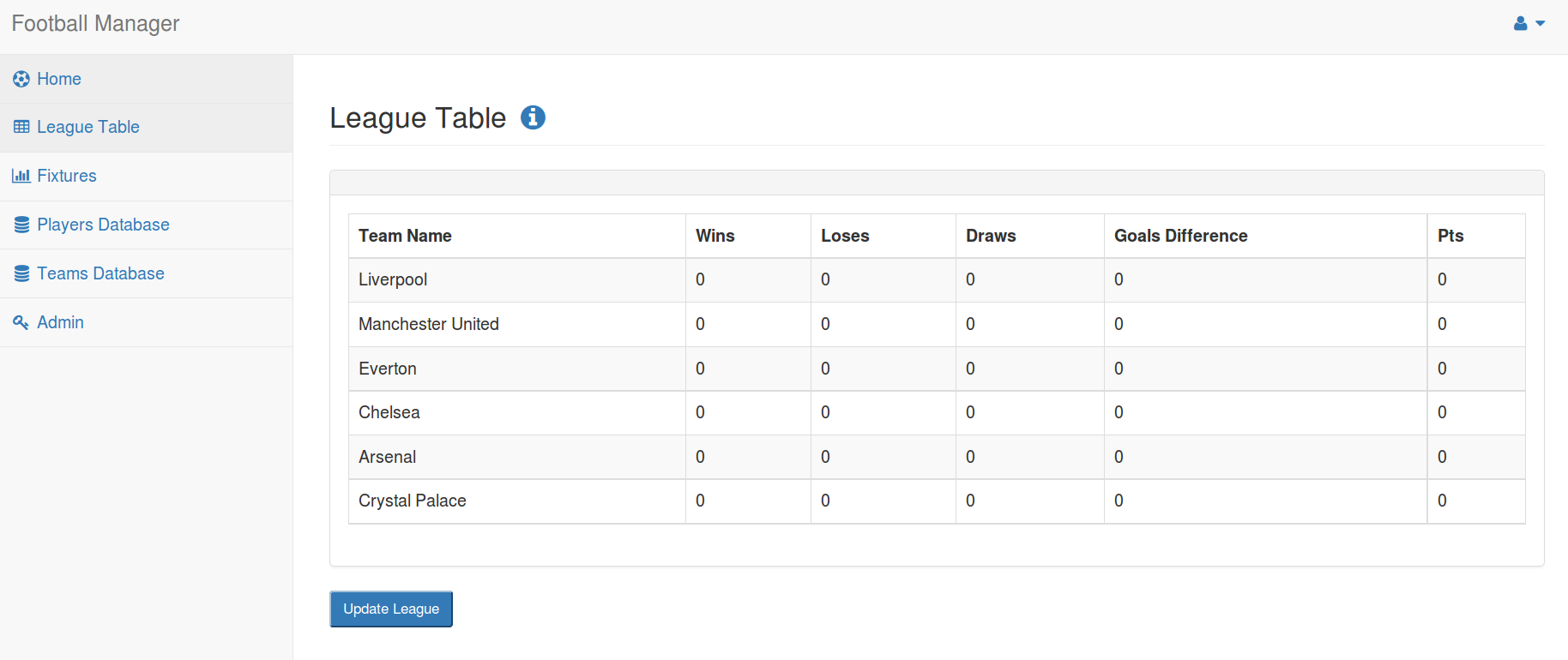
## Fixtures Page



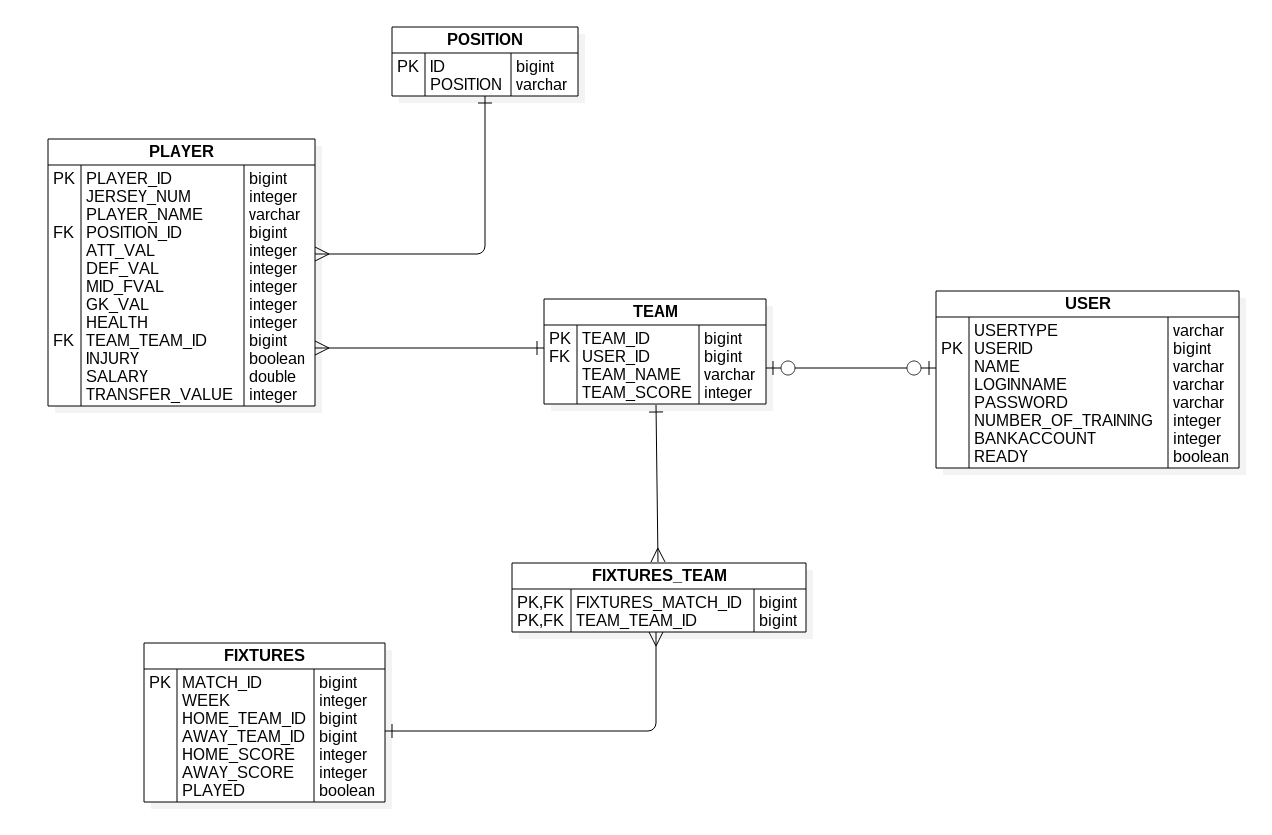
## Squad Page



## League Table



# ERD Diagram



# Table/Record Layout

## Basic System

### User

|  |  |  |
| --- | --- | --- |
| Table: User |  |  |
| Name | Format | Description |
| userID(PK) | VARCHAR2 | User ID at the time of Game |
| name | VARCHAR2 | Persons Name |
| loginName | VARCHAR2 | User login name |
| password | VARCHAR2 | User password |
|  |  |  |
|  |  |  |
| Player |  |  |
| Table: Player |  |  |
| Name | Format | Description |
| playerID(PK) | VARCHAR2 | Players name in the Game |
| teamID(FK) | VARCHAR2 | User ID at the time of Game |
| jerseyNum | VARCHAR2 | Jersey number of team player |
| injury | VARCHAR2 | If player gets injured during the game |
| playerName | VARCHAR2 | Name of a team member |
| position | VARCHAR2 | Football position of the player |
| attVal | NUMBER | How good player is at attack |
| defVal | NUMBER | How good player is at defence |
| gkVal | NUMBER | How good player is at goalkeeping |
| midFVal | NUMBER | How good player is at midfield play |

### Team

|  |  |  |
| --- | --- | --- |
| Table: Team |  |  |
| Name | Format | Description |
| teamID(PK) | VARCHAR2 | Team ID in the League |
| userID(FK) | VARCHAR2 | User ID at the time of Game |
| leagueName(FK) | VARCHAR2 | Name of a League in the Game |
| teamName | VARCHAR2 | Team name in the League |
| teamScore | NUMBER | Team score is total score of players ability |
|  |  |  |
| League |  |  |
| Table: League |  |  |
| Name | Format | Description |
| leagueName(PK) | VARCHAR2 | Name of a League in the Game |
| numOfTeams | NUMBER | Number of a teams in the League |

### Fixtures

|  |  |  |
| --- | --- | --- |
| Table: Fixtures |  |  |
| Name | Format | Description |
| matchID(PK) | VARCHAR2 | Match ID of current fixtures |
| leagueName(FK) | VARCHAR2 | Name of a League in the Game |
| teamID(FK) | VARCHAR2 | Away team ID |
| awayScore | NUMBER | Points scored by away team |
| homeScore | NUMBER | Points scored by home team |
| startTime | NUMBER | Start time of a match |
| endTime | NUMBER | End time of a match |
| played | VARCHAR2 | Indicates if match was already played |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| TeamFixtures |  |  |
|  |  |  |
| Table: TeamFixtures | |  |
| Name | Format | Description |
| fixturesID(PK,FK) | VARCHAR2 | Fixtures ID |
| teamID(PK,FK) | VARCHAR2 | Home team ID |

## Advanced System

### User

|  |  |  |
| --- | --- | --- |
| Table: User |  |  |
| Name | Format | Description |
| userID(PK) | VARCHAR2 | User ID at the time of Game |
| name | VARCHAR2 | Persons Name |
| loginName | VARCHAR2 | User login name |
| password | VARCHAR2 | User password |
|  |  |  |
| Player |  |  |
| Table: Player |  |  |
| Name | Format | Description |
| playerID(PK) | VARCHAR2 | Players name in the Game |
| teamID(FK) | VARCHAR2 | User ID at the time of Game |
| jerseyNum | VARCHAR2 | Jersey number of team player |
| injury | VARCHAR2 | If player gets injured during the game |
| playerName | VARCHAR2 | Name of a team member |
| position | VARCHAR2 | Football position of the player |
| attVal | NUMBER | How good player is at attack |
| defVal | NUMBER | How good player is at defence |
| gkVal | NUMBER | How good player is at goalkeeping |
| midFVal | NUMBER | How good player is at midfield play |

### Team

|  |  |  |  |
| --- | --- | --- | --- |
| Table: Team |  | |  |
| Name | Format | | Description |
| teamID(PK) | VARCHAR2 | | Team ID in the League |
| userID(FK) | VARCHAR2 | | User ID at the time of Game |
| leagueName(FK) | VARCHAR2 | | Name of a League in the Game |
| teamName | VARCHAR2 | | Team name in the League |
| teamScore | NUMBER | | Team score is total score of players ability |
|  |  | |  |
| League |  | |  |
| Table: League | | | |
| Name | Format | | Description |
| leagueName(PK) | VARCHAR2 | | Name of a League in the Game |
| numOfTeams | NUMBER | | Number of a teams in the League |
| Fixtures |  | |  |
| Table: Fixtures | | | |
| Name | Format | | Description |
| matchID(PK) | VARCHAR2 | | Match ID of current fixtures |
| leagueName(FK) | VARCHAR2 | | Name of a League in the Game |
| teamID(FK) | VARCHAR2 | | Away team ID |
| awayScore | NUMBER | | Points scored by away team |
| homeScore | NUMBER | | Points scored by home team |
| startTime | NUMBER | | Start time of a match |
| endTime | NUMBER | | End time of a match |
| played | VARCHAR2 | | Indicates if match was already played |
| ticketsSold | NUMBER | | number of tickets sold for the match |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
| TeamFixtures |  | |  |
| Table: TeamFixtures | | |  |
| Name | | Format | Description |
| fixturesID(PK,FK) | | VARCHAR2 | Fixtures ID |
| teamID(PK,FK) | | VARCHAR2 | Home team ID |
|  | |  |  |
|  | |  |  |

### Stadium

|  |  |  |
| --- | --- | --- |
| Table: Stadium |  |  |
| Name | Format | Description |
| stadID(PK) | VARCHAR2 | ID of the Stadium in use |
| teamID(FK) | VARCHAR2 | ID of the team using the stadium |
| numOfSeats | NUMBER | capactity of the stadium measured in seats |
| ticketCost | NUMBER | cost of a ticket to the stadium |
| stadType | VARCHAR2 | type of stadium |
|  |  |  |
| Bank |  |  |
| Table: Bank |  |  |
| Name | Format | Description |
| accNum(PK) | VARCHAR2 | account number to for the specific user |
| userID(FK) | VARCHAR2 | users ID associated with the user |
| balance | NUMBER | balance in the account |
|  |  |  |
|  |  |  |
|  |  |  |

# Test Case Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case Matrix | | | | | |
| Test Case No | Test Case Name | Input | Expected Outcome | Actual Outcome | Result Pass/Fail |
| 1 | Register user | Phill : name [phill.cheape@gmail.com](mailto:phill.cheape@gmail.com):email  Pcheape: username, password:Password  password: confirm password.  Team1:Team name | Account created |  |  |
| 2 | Register user password not confirmed | Phill : name [phill.cheape@gmail.com](mailto:phill.cheape@gmail.com):email  Pcheape: username, password:Password  pass: confirm password.  Team1:Team name | Passwords do not match |  |  |
| 3 | User name Taken | Phill : name [phill.cheape@gmail.com](mailto:phill.cheape@gmail.com):email  Pcheape: username, password:Password  password: confirm password.  Team1:Team name | User name taken error |  |  |
| 4 | Field left blank | : name [phill.cheape@gmail.com](mailto:phill.cheape@gmail.com):email  Pcheape: username, password:Password  password: confirm password.  Team1:Team name | Unable to submit |  |  |
| 5 | Email incorrect format | Phill : name [phill.cheapegmail.com](mailto:phill.cheape@gmail.com):email  Pcheape: username, password:Password  password: confirm password.  Team1:Team name | Incorrect email error |  |  |
|  | Login | Pcheape:username  password:Password | Login |  |  |
|  | Login incorrect password | Pcheape:username  pass:Password | Incorrect username/password |  |  |
|  | Login incorrect username | Pche:username  password:Password | Incorrect username/password |  |  |
|  | Manage team | Setup 11 players + 3 subs submit | Team submitted |  |  |
|  | Manage team | Setup 10 players + 3 subs submit | Incorrect Team please choose 11 players |  |  |
|  | Admin edit user | Pnill | Username changed |  |  |
|  | Team edit | Team2:Team name | Team name changed |  |  |
|  | Admin edit user existing name | pcheape | User already exists |  |  |
|  | Play match win | Win | +3 points on league and associated stats |  |  |
|  |  |  |  |  |  |
|  | Play match lose | Lose | +0points on league table and associated stats |  |  |
|  | Play match draw | draw | +1 points on league table and associated stats |  |  |
|  | Train Player fail | Gk | No change on stat |  |  |
|  | Train Player Pass | Gk | +1 to goalKeeping stat |  |  |
|  | Player injured | (these are random so will have to keep playing and training player till he is injured ) | Player injured. |  |  |
|  | Injured player in team | Setup 10 + players plus injured player | Incorrect Team please choose 11 players |  |  |

# Pseudo Code

## calculateMatch()

|  |  |
| --- | --- |
|  | |
| **Pseudo Code** | calculateMatch() |
|  |  |
|  | **Inputs:** |
|  | **Name: Type** |
|  | teamaScore Int |
|  | teambScore Int |
|  |  |
|  |  |
|  |  |
|  | **Outputs:** |
|  | **Name: Type** |
|  | **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** |
|  |  |
|  |  |
| **Algorithm** | **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 |
|  | then 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 |
|  | end if |
|  | begin if |
| rounded up | goalKeeperScore - attackScore > 0 |
|  | then LoserScore = (goalKepperScore-attackScore)/2 |
|  | end if |
|  | calculateResults(getFixtureID(),teamAScore int, teamBScore int) \\team A will always be the home team |
|  | **End** |

## addTrainVal()

|  |  |
| --- | --- |
| **Pseudo Code** | addTrainVal() |
|  |  |
|  | **Inputs:** |
|  | **Name: Type** |
|  | trainingVal int |
|  | playerID int |
|  | positionString |
|  |  |
|  |  |
|  | **Outputs:** |
|  | **Name: Type** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | **Constants** |
|  | **Name: Type** |
|  | MAX\_VAL int |
|  |  |
|  |  |
|  | **Other** |
|  | **Name: Type** |
|  | overByMaxVal int |
|  | playerVal int |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Algorithm** | **Begin** |
|  | playerVal = getPlayerPositionVal(playerID int, positon String) |
|  | begin if |
|  | playerVal + trainVal <= MAX\_VAL |
|  | then playerVal = playerVal + trainVal |
|  | else |
|  | begin if |
|  | playerVal + trainVal > MAX\_VAL |
|  | then overByMaxVal = (playerVal + trainVal) - maxVal |
|  | end if |
|  | playerVal = (playerVal + trainVal) - overByVal |
|  | end if |
|  | setPlayerStatistic(playerID int, position String, playerVal int) |
|  | **End** |

## calculateResults()

|  |  |  |
| --- | --- | --- |
| **Pseudo Code** | calculateResults() |  |
|  |  |  |
|  | **Inputs:** |  |
|  | **Name: Type** |  |
|  | homeScore int |
|  | awayScore int |  |
|  | fixtureID int |  |
|  | teamA |  |
|  | homeTeamID int |  |
|  | awayTeamID int |  |
|  | **Outputs:** |  |
|  | **Name: Type** |  |
|  |  |  |
|  |  |  |
|  | **Constants** |  |
|  | **Name: Type** |  |
|  | WIN\_POINTS int |  |
|  | DRAW\_POINTS int |  |
|  |  |  |
|  | **Other** |  |
|  | **Name: Type** |  |
|  | teamAGoals int |  |
|  | winingTeam char |  |
|  | teamBGoals int |  |
|  | teamAPoints int |  |
|  | teamBPoints int |  |
|  |  |  |
| **Algorithm** | **Begin** |  |
|  | teamAGoals = homeScore |  |
|  | TeamBGoals = awayScore |  |
|  | begin if |  |
|  | homeScore > awayScore |  |
|  | then |  |
| \\home team always is team A | winningTeam = A |  |
|  | teamAPoints = WIN\_POINTS |  |
|  | teamBPoints = 0 |  |
|  | else if |  |
|  | homeScore == awayScore |  |
|  | teamAPoints = DRAW\_POINTS |  |
|  | teamBPoints = DRAW\_POINTS |  |
|  | else |  |
|  | then |  |
|  | winningTeam = B |  |
|  | teamBPoints = WIN\_POINTS |  |
|  | teamAPoints = 0 |  |
|  | end if |  |
|  | END |  |
|  |  |  |

## getTrained()

|  |  |
| --- | --- |
| **Pseudo Code** | getTrained() |
|  |  |
|  | **Inputs:** |
|  | **Name: Type** |
|  | position String |
|  | playerId int |
|  |  |
|  |  |
|  |  |
|  | **Outputs:** |
|  | **Name: Type** |
|  | Trained boolean |
|  |  |
|  |  |
|  |  |
|  |  |
|  | **Constants** |
|  | **Name: Type** |
|  | injuryHealth int |
|  | maxPlayerStat int |
|  |  |
|  | **Other** |
|  | **Name: Type** |
|  | randomTrainVal (1-5) |
|  | randomHealthLose(1-4) |
|  | health int |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Algorithm** | **Begin** |
|  | generate randomTrainVal(1-5) |
|  | Begin if |
|  | playerMaxed(playerID) == true |
|  | then handle maxedPlayerException and retrun false |
|  | else if |
|  | randomTrainVal <=2 |
|  | then return false |
|  | else if |
|  | randomTrainVal = 3 or randomTrainVal = 4 |
|  | then generate randomHealthLose(1-4) - 1 |
|  | addTrainVal(randomTrainVal int, playerID int, position String) |
|  | health = getPlayerHealth(playerID int - randomHealthLose |
|  | getInjured(heath int, playerId int) |
|  | return true |
|  | else |
|  | then generate randomHealthLose(3-4) |
|  | addTrainVal(randomTrainVal int, playerID int, position String) |
|  | health = getPlayerHealth(playerID int - randomHealthLose |
|  | getInjured(heath int, playerId int) |
|  | end if |
|  | **End** |

## manageTeam()

|  |  |  |
| --- | --- | --- |
| **Pseudo Code** | manageTeam() |  |
|  |  |  |
|  | **Inputs:** |  |
|  | **Name: Type** |  |
|  | newTeamName String |  |
|  | newPlayerPosition String |  |
|  | newPlayerID int |  |
|  | oldPlayerID int |  |
|  | oldPlayerPosition String |  |
|  | **Outputs:** |  |
|  | **Name: Type** |  |
|  | successful boolean |  |
|  |  |  |
|  | **Constants** |  |
|  | **Name: Type** |  |
|  | MAX\_PLAYERS int |  |
|  | MAX\_ON\_FIELD int |  |
|  | MAX\_SUBS int |  |
|  |  |  |
|  | **Other** |  |
|  | **Name: Type** |  |
|  |  |  |
|  |  |  |
| **Algorithm** | **Begin** |  |
|  | pass newTeamName to database |  |
|  | set oldPlayerID as newPlayerID |  |
|  | set oldPlayerPosition as newPlayerPosition |  |
|  | begin if |  |
|  | teamSize > MAX\_PLAYERS |  |
|  | then return false |  |
|  | else if |  |
|  | numberSubs > MAX\_SUBS |  |
|  | then return false |  |
|  | else if |  |
|  | numberOnField > MAX\_ON\_FIELD |  |
|  | then return false |  |
|  | else |  |
|  | commit changes |  |
|  | return true |  |
|  | end if |  |
|  | **End** |  |

# User Manual