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](#_Toc449359548)

[Introduction 4](#_Toc449359549)

[User Requirements Document 5](#_Toc449359550)

[Introduction 5](#_Toc449359551)

[Survey 6](#_Toc449359552)

[Results 7](#_Toc449359553)

[Requirements 9](#_Toc449359554)

[System Narrative 10](#_Toc449359555)

[Use Case Diagram 11](#_Toc449359556)

[Use Case Description 12](#_Toc449359557)

[Login 12](#_Toc449359558)

[Register 16](#_Toc449359559)

[Manage Team 20](#_Toc449359560)

[Play Match 24](#_Toc449359561)

[Edit Team 28](#_Toc449359562)

[Class Diagram 32](#_Toc449359563)

[Design Specification 33](#_Toc449359564)

[Website Story Board 34](#_Toc449359565)

[Website Screenshots 36](#_Toc449359566)

[Home Page 36](#_Toc449359567)

[Login Page 37](#_Toc449359568)

[Register Page 38](#_Toc449359569)

[Fixtures Page 39](#_Toc449359570)

[Squad Page 40](#_Toc449359571)

[League Table 41](#_Toc449359572)

[ERD Diagram 42](#_Toc449359573)

[Table/Record Layout 43](#_Toc449359574)

[Basic System 43](#_Toc449359575)

[User 43](#_Toc449359576)

[Player 43](#_Toc449359577)

[Team 44](#_Toc449359578)

[League 44](#_Toc449359579)

[Fixtures 45](#_Toc449359580)

[TeamFixtures 45](#_Toc449359581)

[Advanced System 46](#_Toc449359582)

[User 46](#_Toc449359583)

[Player 46](#_Toc449359584)

[Team 47](#_Toc449359585)

[League 47](#_Toc449359586)

[Fixtures 48](#_Toc449359587)

[TeamFixtures 49](#_Toc449359588)

[Stadium 50](#_Toc449359589)

[Bank 50](#_Toc449359590)

[Test Case Matrix 51](#_Toc449359591)

[Pseudo Code 54](#_Toc449359592)

[calculateMatch() 54](#_Toc449359593)

[addTrainVal() 56](#_Toc449359594)

[calculateResults() 58](#_Toc449359595)

[getTrained() 60](#_Toc449359596)

[manageTeam() 62](#_Toc449359597)

[User Manual 64](#_Toc449359598)

[Conclusion 64](#_Toc449359599)

[References 64](#_Toc449359600)

# Introduction

This project is about online simulator of football manager game.

Football is the most popular game in the world, with a large market in pc gaming . Our aim was to engage the large football player base from around the world. And we tried to make this game interesting and exciting for all fans of football.

Once in, you take charge of your own team, buy and sell players, train your players to improve in the areas you wish to focus on, play matches and ultimately win the league.

Football Manager is the most action-packed football manager game of them all, with 1 league match and 3 training sessions per week. Express your management prowess and become the best.

From the above description the audience for this football simulator is intended for all genders and ages.

# 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. For the base system we settled on must haves for the game this included the ability to select a team, play matches, train players, Set positions , and have player injuries along with the admin functions of create/delete/update of players, teams and users .

We then expanded on this with a basic economy and the ability to buy and sell players and give managers the ability to submit teams to create an automated system environment as part of the automation the league table updates after a match is played to give up to date

Information on how the league is progressing . As well as the automatic scheduling of fixtures so no two teams play on the same week.

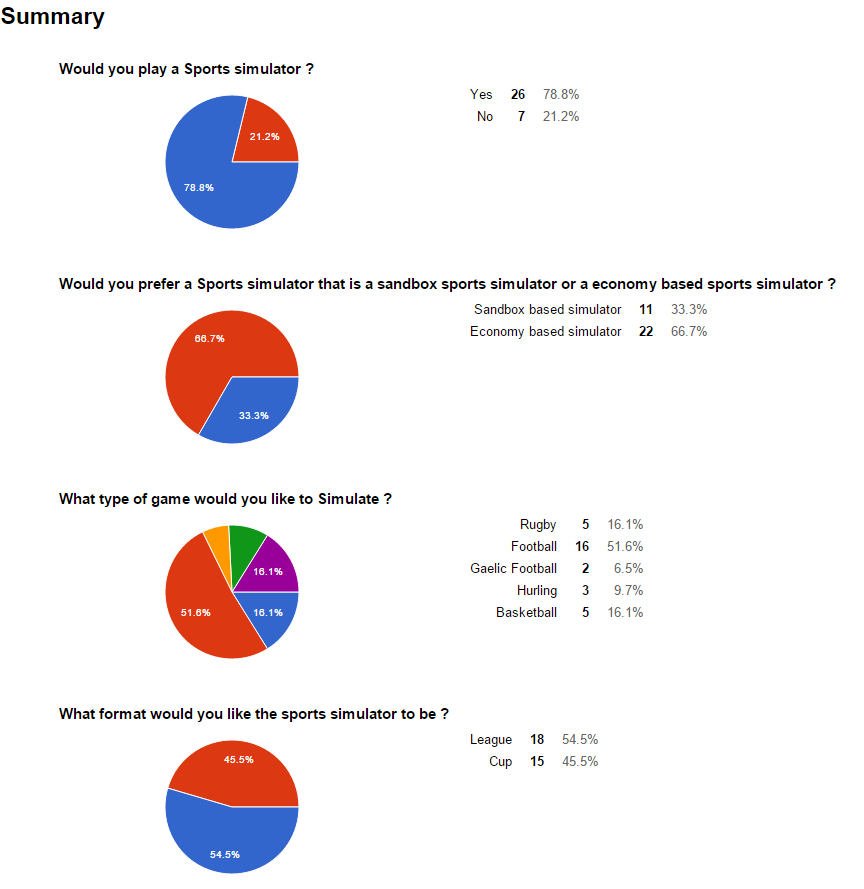
We also focused on the security of the system and gave sha-256 encryption with all passwords which also secures them with a salt value. The algorithm for sha-256 was provided by Yong (2012).

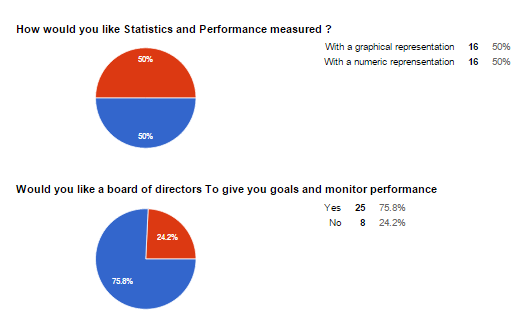
## Survey

* 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.

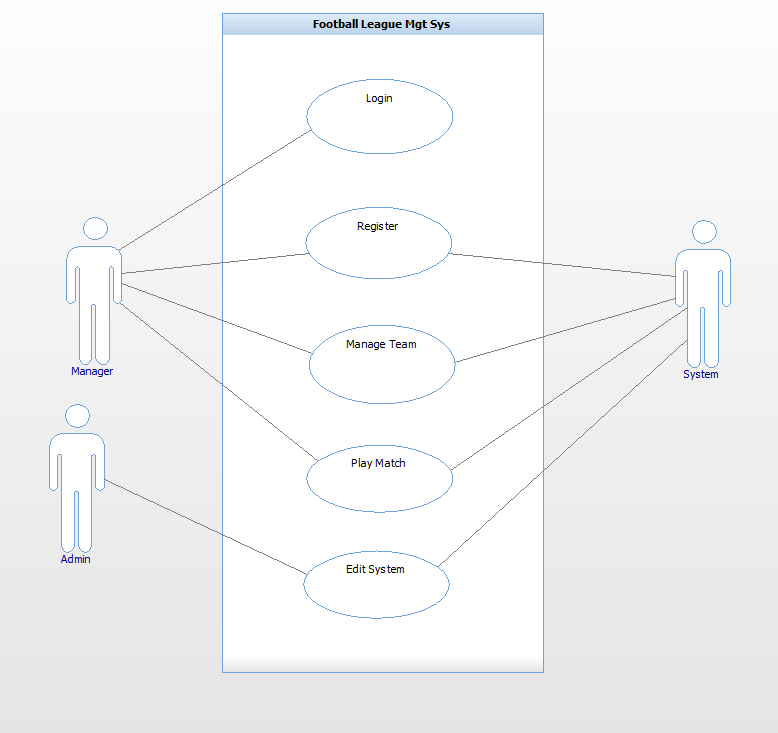
In the Editor the Users will be able to change the player names and team name at any stage throughout the league.

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. 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. |
| 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. |
| 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 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. |
| Use Case: | Manage Team |
| Actor(s): | Manager, System |
| Goal: | To manage their own team. |
| Overview: | It's allow Manager Change team name, Training - to increase a player’s performance , Assign players to positions , Submit for match, Buy and sell players. |
| 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. |

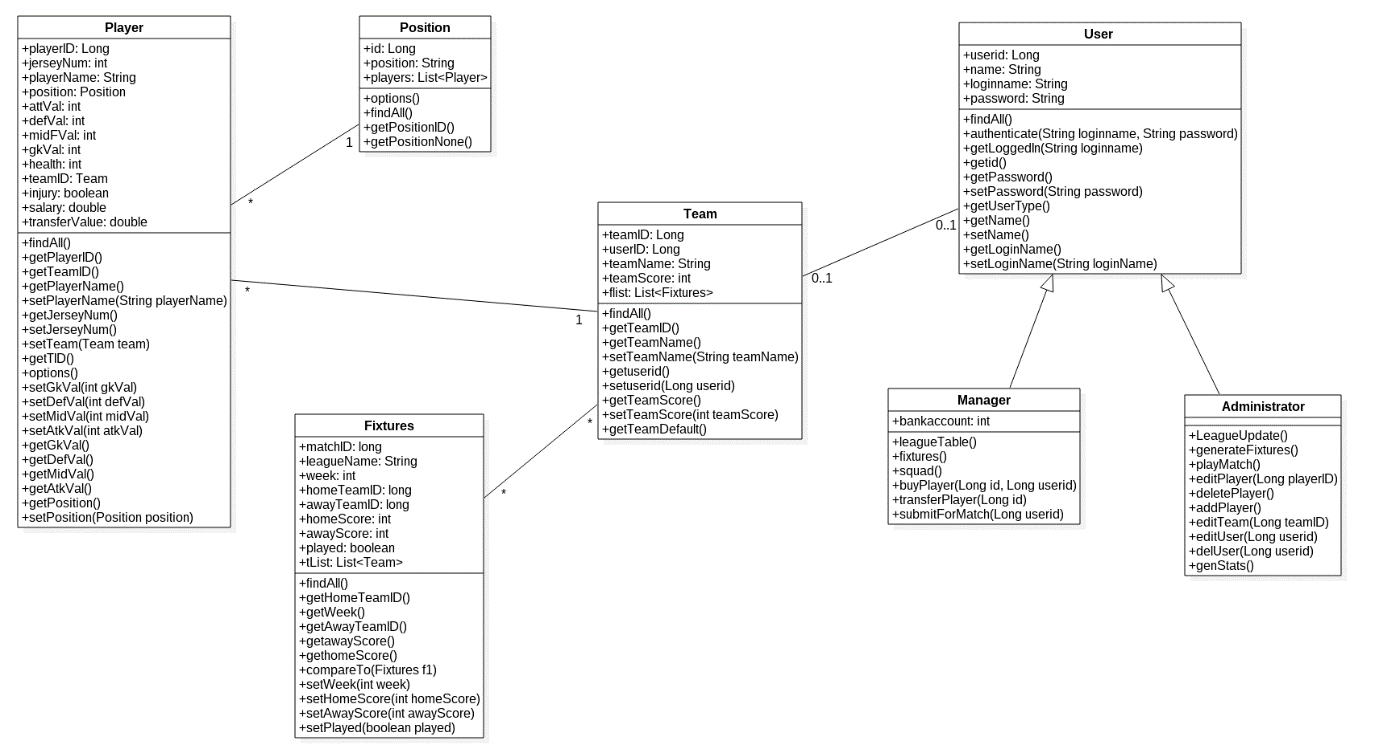
## Play Match

|  |  |
| --- | --- |
| 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. |
| Use Case: | Play Match |
| Actor(s): | Manager, System |
| Goal: | To successfully play match and get results of the match. |
| 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. User sets his team and presses submit. |
| Post-  Condition: | 1. Successfully plays the match, the system provides the outcome of the game. |
| Successful  Scenario: | 1. Checks all managers are ready .  2. Calculates weeks matches.  3. updates League table and fixtures. |
| Alternative  Scenario(s): | 1. If other managers not ready This message is displayed |

## 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 |
| Use Case: | Edit Team |
| Actor(s): | Admin |
| Goal: | To successfully Edit team, change name, Change user. |
| Overview: | This allows the admin to edit team name and details. |
| Pre-  Condition: | N/A |
| Post-  Condition: | N/A |
| Successful  Scenario: | 1. Administrator logs in  2. Chooses team name  3. assigns user to team |
| Alternative  Scenario(s): | 1. If same name as other team displays error. 2. If user already assigned displays error. |

# Class Diagram



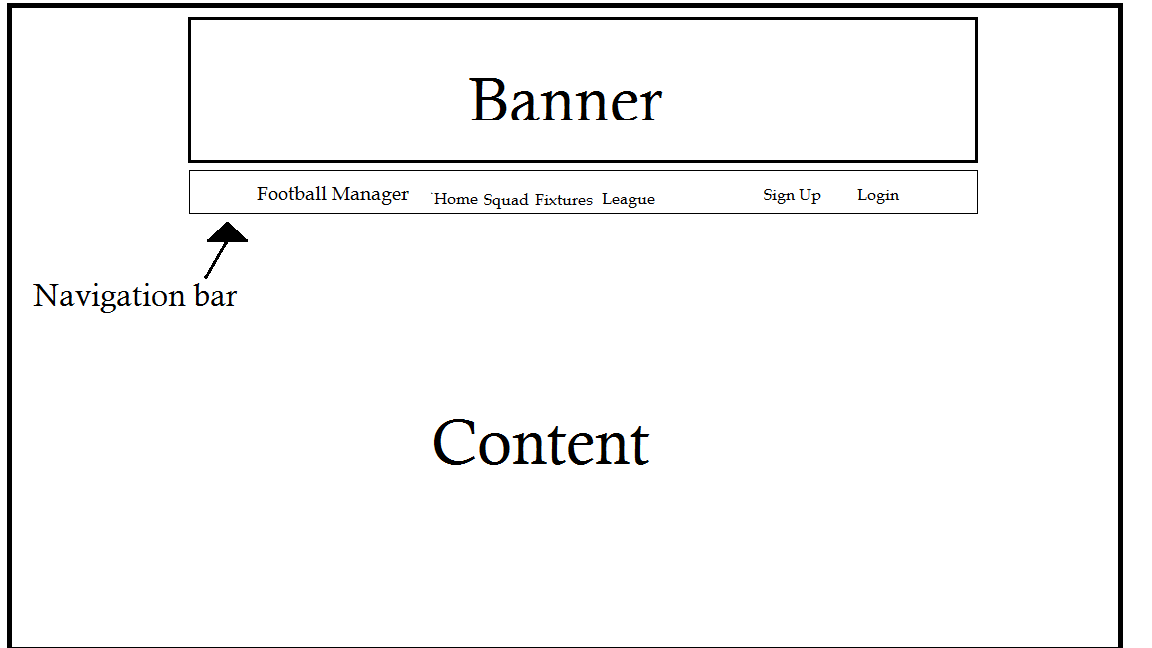
# Design Specification

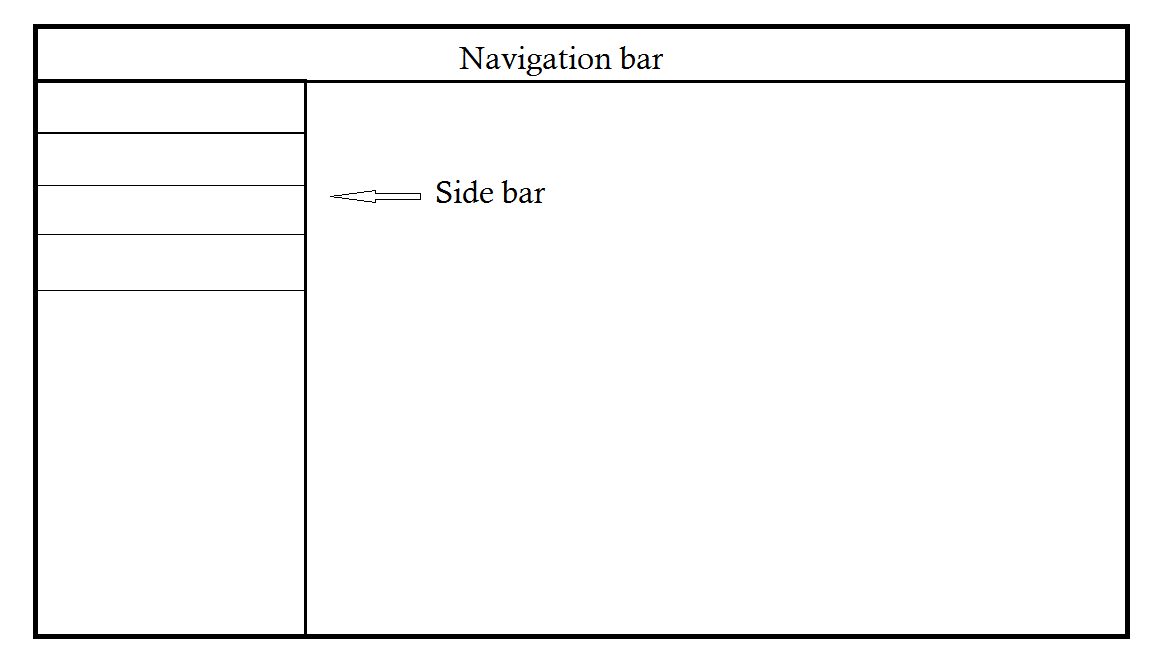
NOTES:

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

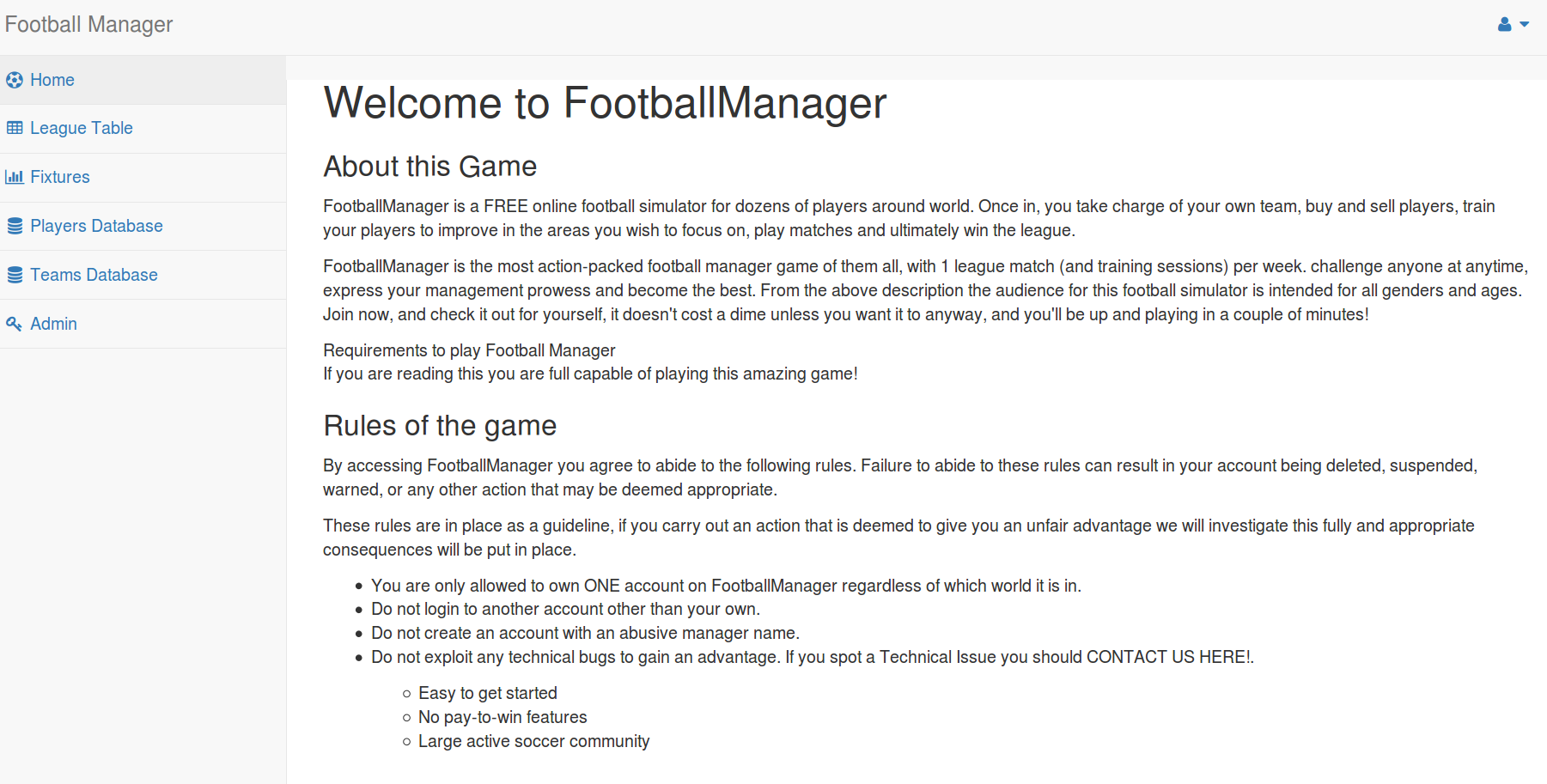
These are the patterns in which we will base the design of our website. The reason why this was chose is due to its simplicity and clarity to present the contents which will be displayed in our website. The template used was taken from www.startbootstrap.com (see pg. 64).



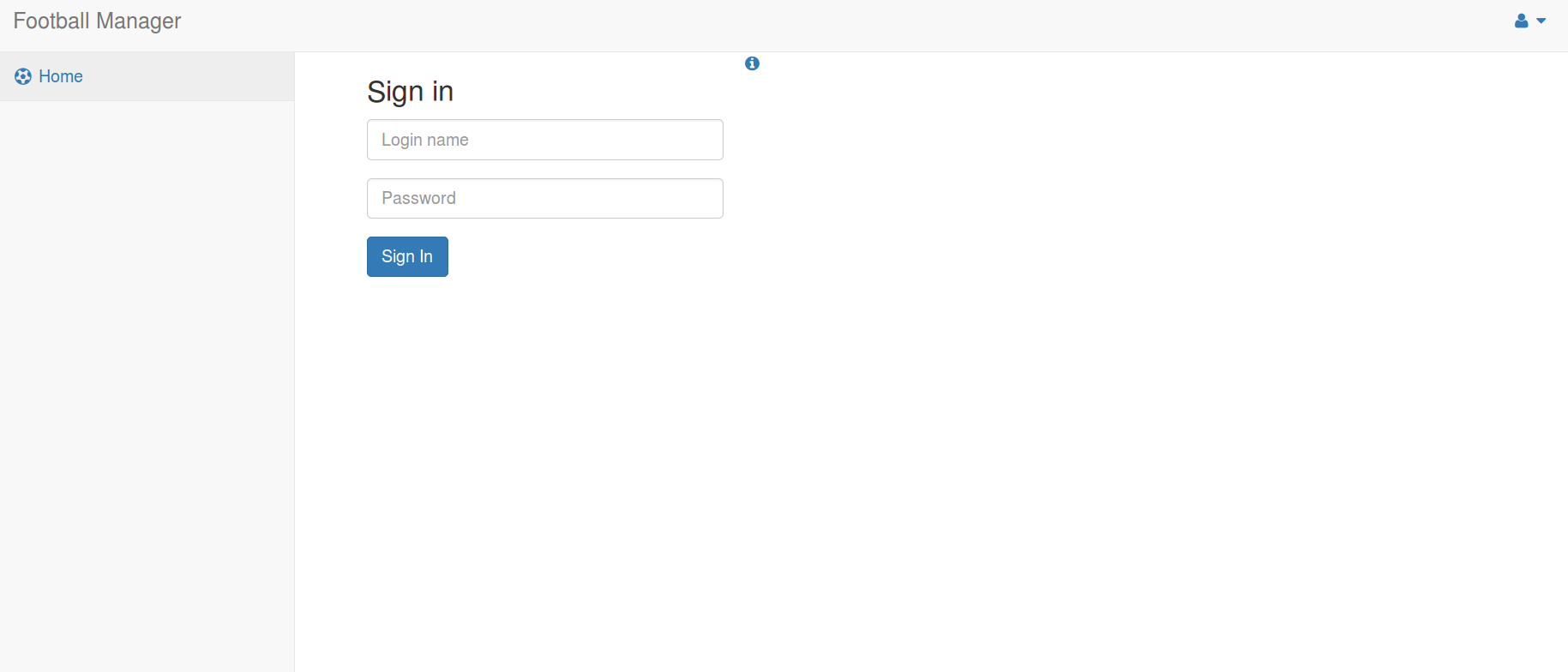


# Website Screenshots

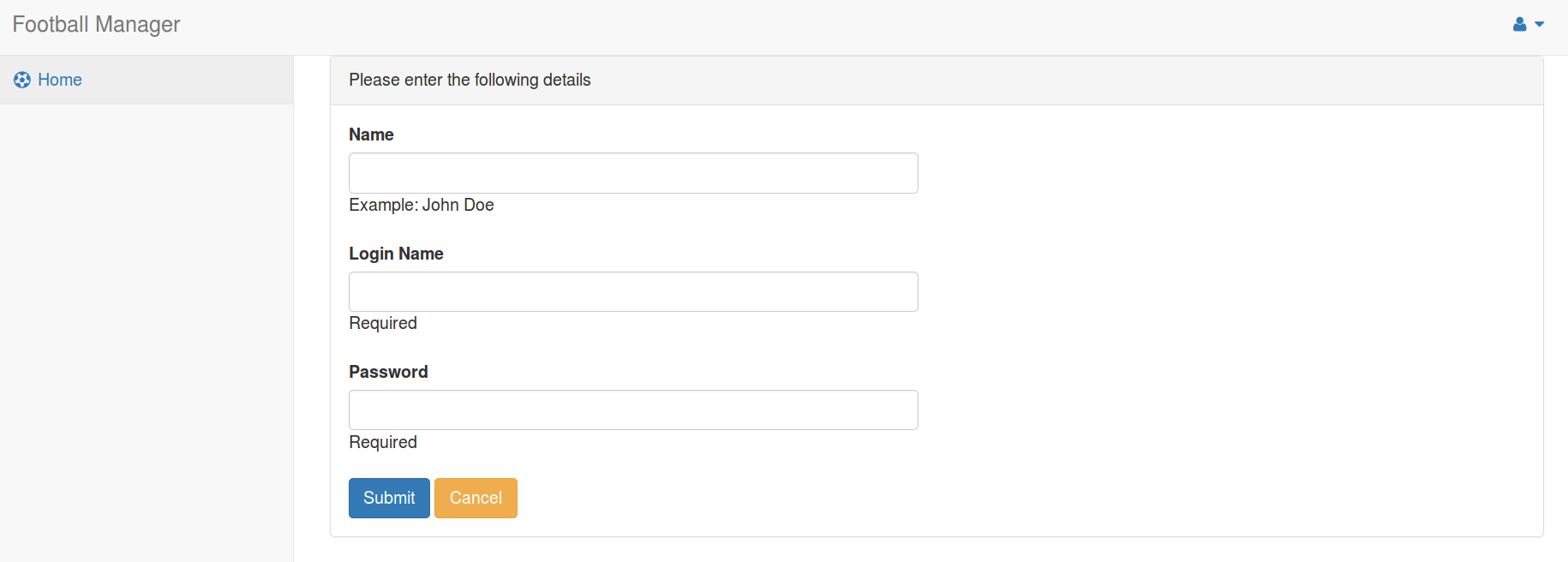
## Home Page



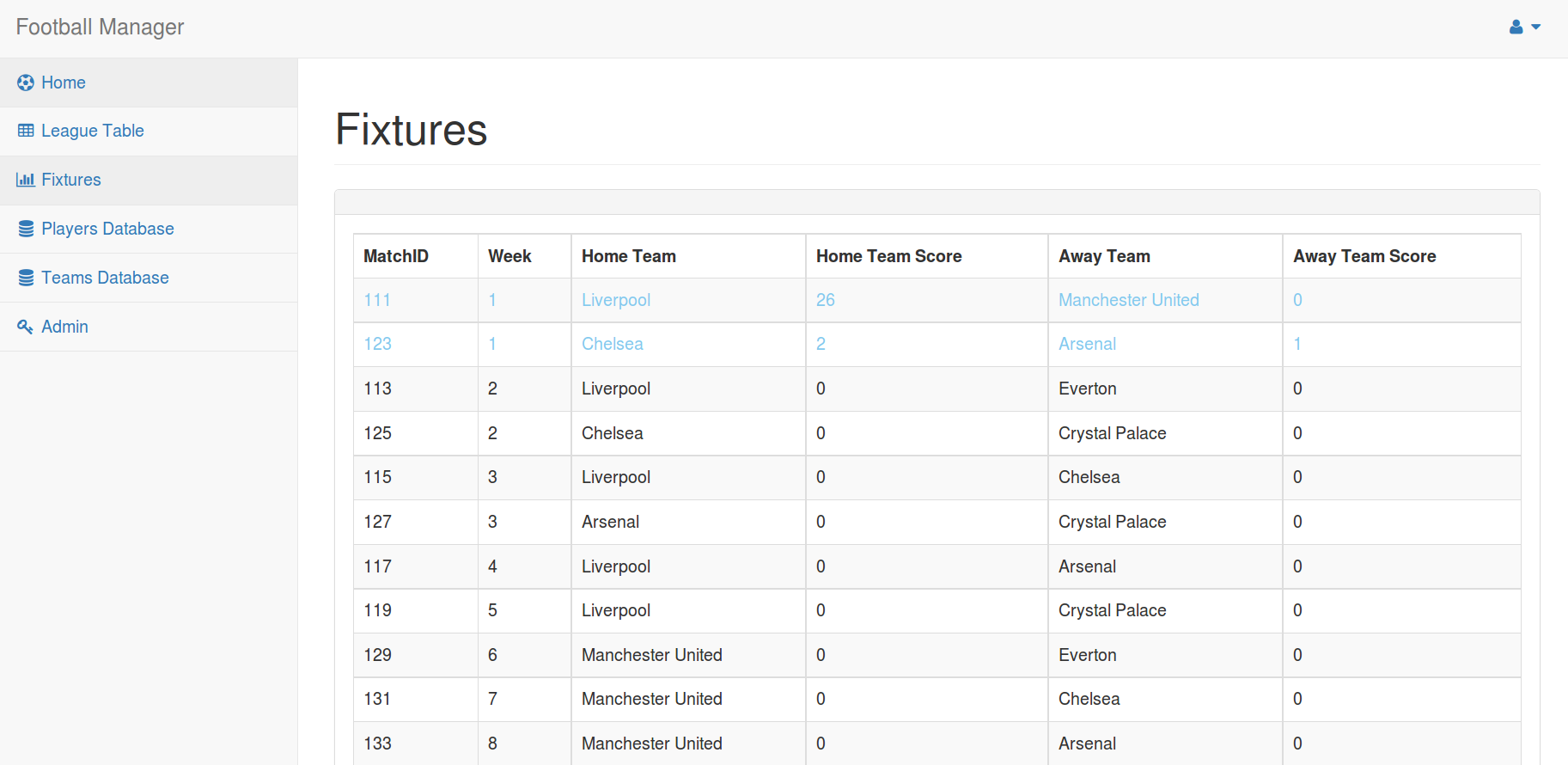
## Login Page



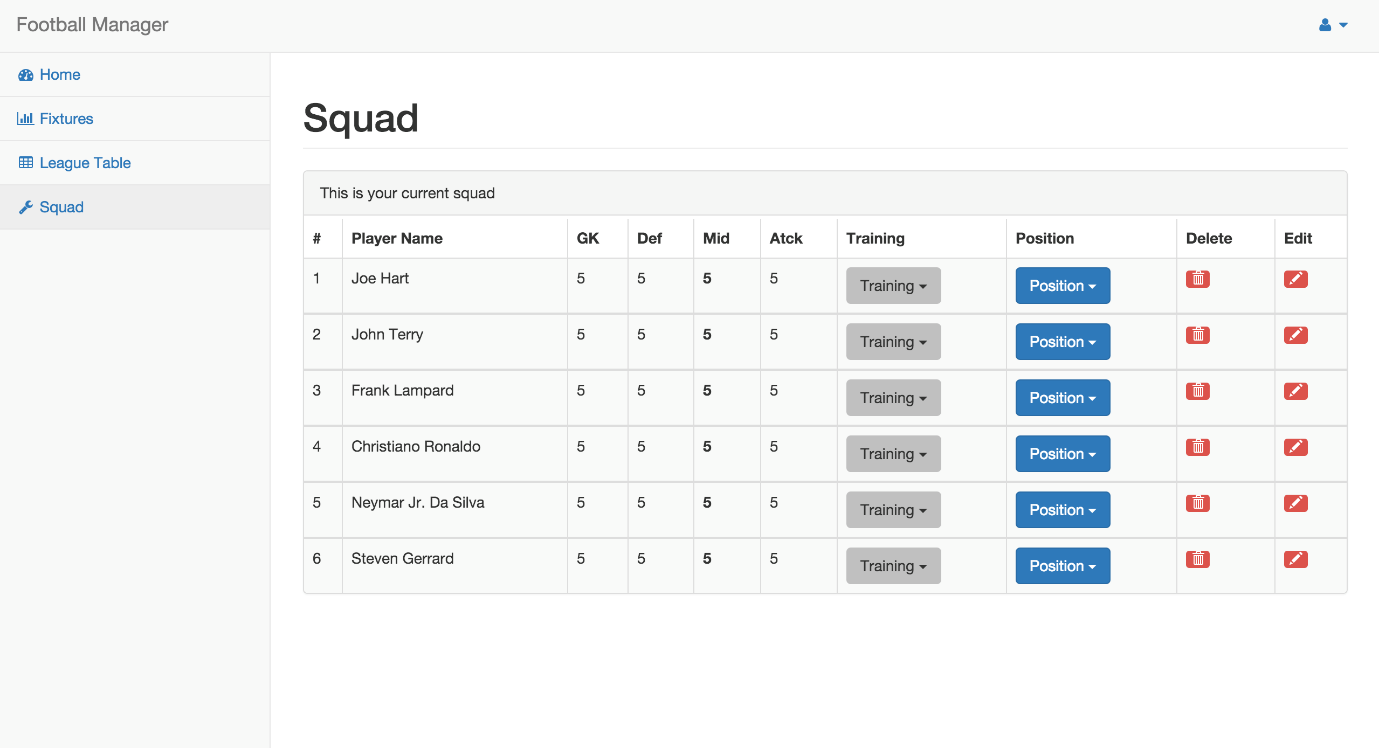
## Register Page



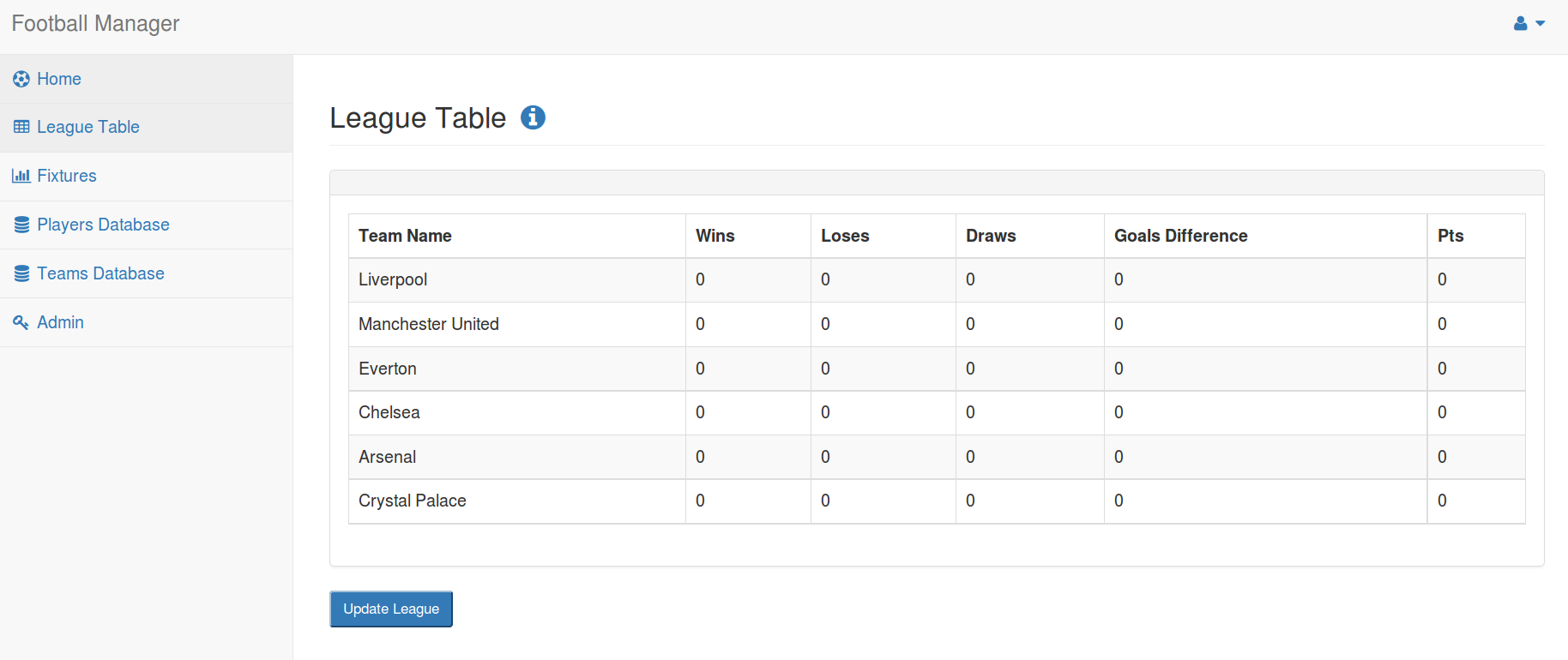
## Fixtures Page



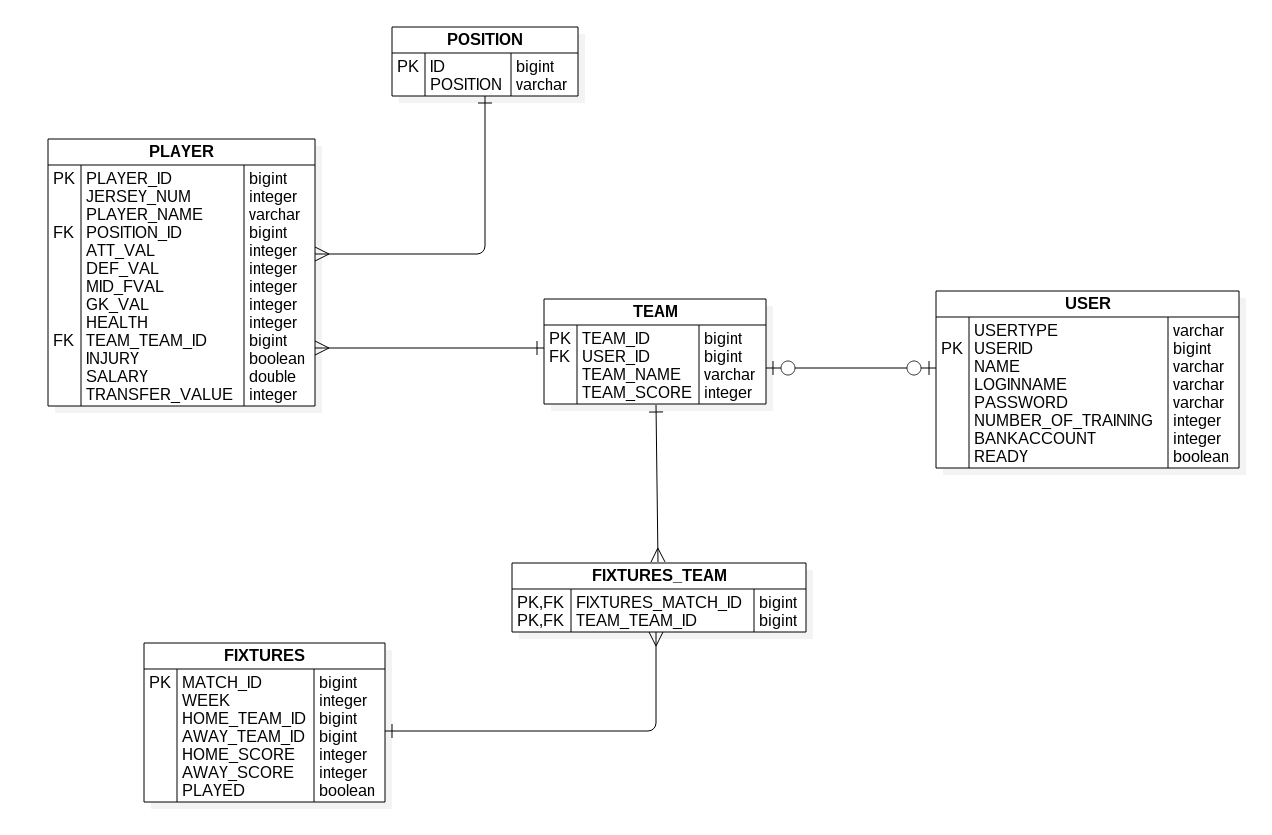
## Squad Page



## 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 No** | **Test Case Name** | **Input** | **Expected Outcome** | **Actual Outcome** | **Result (Pass/Fail)** |
| 1 | Register user | name: phill login name phill password phill | login redirect | login redirect | pass |
| 2 | User name Taken | name: phill login name phill password phill | Login name is already used please try again! | Login name is already used please try again! | pass |
| 3 | Field left blank | name: phill login name password phill | This field is required | This field is required | pass |
| 4 | Login success | loginname phill password phill | redirect to home | redirect to home | pass |
| 5 | Login incorrect password | loginname phill password | Invalid user or password | Invalid user or password | pass |
| 6 | Login incorrect username | loginname password phill | Invalid user or password | Invalid user or password | pass |
| 7 | Manager select team | 1 goalkeeper 4 def 4 attacks 4 subs. | Team selected | teamSelected | pass |
|  | Submit team 2 managers | submit | awaiting managers | blank | fail |
| 8 | Submit team 2 managers | submit | match played | match played | pass |
| 10 | too many defenders | 5\*defenders | sorry you already have enough players in position defender | sorry you already have enough players in position defender | pass |
| 11 | too many midfield | 5\*midfield | sorry you already have enough players in position midfield | sorry you already have enough players in position midfield | pass |
| 12 | too many attackers | 3\*attackers | sorry you already have enough players in position striker | sorry you already have enough players in position striker | pass |
| 13 | too many goalkeepers | 2\*goalkeepers | sorry you already have enough players in position GoalKeeper | sorry you already have enough players in position GoalKeeper | pass |
| 14 | set player goalKeeper | 1\*goalkeeper | player positon = position | player positon = position | pass |
| 15 | set player midfield | 1\*midfield | player positon = position | player positon = position | pass |
| 16 | set player striker | 1\*striker | player positon = position | player positon = position | pass |
| 17 | set player not assigned | 1\*notassigned | player positon = position | player positon = position | pass |
| 18 | buy player | buy player from transfer market | bankaccount+value | bankaccount is too much | pass |
| 19 | sell player |  | bankaccount-value | bankaccount-value | Pass |
| 20 | Change Team name | Celtic | name changed | name changed | pass |
| 21 | Team already Exists | everton | Team already Exists | Team already Exists | pass |
| 22 | Generate fixtures |  | (fixtures generated with no two teams playing same week) | (fixtures generated with no two teams playing same week) | pass |
| 23 | play weeks match . |  | week matches played | week matches played | pass |
| 24 | League win |  | +3 points | +3 points | pass |
| 25 | League loss |  | 0 points | 0 points | pass |
| 26 | League Draw |  | +1 | 1 | pass |
| 27 | Team filter none |  | displays all position none players | displays all | pass |
| 28 | team filter goalKeeper |  | displays all position goalKeeper players | displays all position goalKeeper players | pass |
| 29 | team filter defender |  | displays all position Defender players | displays all position Defender players | pass |
| 30 | team filter striker |  | displays all position striker players | displays all position striker players | pass |
| 31 | team filter sub |  |  |  |  |
| 32 | team filter sub |  | displays all position sub players | displays all position sub players | pass |
| 33 | Train player | train player first time | goalkeeper chance of stat going up . | goalkeeper chance of stat going up . | pass |
| 34 | Train player | train same player | Sorry could not train player this time please try again | Sorry could not train player this time please try again | pass |
| 35 | delete player | delete craig hill | player deleted | player deleted | pass |
| 36 | edit player playerdb | 400 atk | stat must be under 10 | stat must be under 11 | pass |
| 37 | edit player playerdb | 401 mid | stat must be under 10 | stat must be under 12 | pass |
| 38 | edit player playerdb | 402 def | stat must be under 10 | stat must be under 13 | pass |
| 39 | edit player playerdb | 403 gk | stat must be under 10 | stat must be under 14 | pass |
| 40 | edit player playerdb | 400 health | stat must be under 10 | stat must be under 15 | fail |
| 41 | edit player playerdb | change name to phillip | name changed | name changed | pass |
| 42 | Change Team name | change name hearts | name changed | name changed | pass |
| 43 | change team name to name that exists | change name to everton | sorry team already exists | name changed | pass |
| 44 | add player | add bing | player bing has been added | player bing has been added | pass |
| 45 | assign team to user | assign everton to user 1 | Team Everton has been updated | Team Everton has been updated | pass |
| 46 | assign team to same user | assign manchester to user1 | sorry user already has team | Team Manchester United has been updated | pass |
| 47 | add user | add bing as admin | Admin bing has been registered | Admin bing has been registered | pass |
| 48 | add same user | add bing as admin | username taken | Admin bing has been registered | pass |
| 49 | update password | change password to pass | User bing has been updated | User bing has been updated | pass |
| 50 | change name | name bing changed to graham | User bings has been updated | User bings has been updated | pass |
| 51 | change name to same | name bing changed to admin | username taken | username taken | pass |
| 52 | delete user | delete user bing | user deleted no flash | user deleted no flash | pass |
| 53 | add team | add team everton | team name already in use | team name already in use | pass |
| 54 | add team | no | Team no has been created | Team no has been created | pass |
| 55 | Generate fixtures uneven teams | generate fixuters with uneven teams | Odd number of teams please add a team | Odd number of teams please add a team | pass |
| 56 | injured player assigned to team | assign player to goalkeeper | player is injured cant assign to team | Player has changed position to Goalkeeper! | fail |

# 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

The reason why the User Manual is not included in this document is due to the fact that our website is an online system, the user manual is located on the website under the page help.

# Conclusion

We were prompt from the start to make an online simulator, this documentation describes the process and stages in which we have partaken to complete this task.

We started by surveying a score of students in order to to figure out which features would be necessary to include in this simulator. From the results gathered from the survey we discovered that many of the percipients of the survey were willing to play a football simulator with an economy and it must a league type of format.

We then proceeded to complete the requirements of this project such as; use case diagram, use case description, class diagram, website story board, website screenshots, ERD Diagram, Table/Record layout, Test Case Matrix and pseudo code for our main functionalities. These steps helped us in the overall development of this simulator.

For the coding we each learned how to use Git. A directory for our project was created on Github and each member then had a separate branch, when each individual’s code was compiling and running we each merged the code to the master branch. Using Git was a significant help and a vital part for collaboration and code sharing.

There are many features in which we could’ve added to this but due to time we have concluded with these functions. The overall outcome of this project we are satisfied.

During this project we’ve learned how to work on a team, organise meetings and meet the required deadlines.

# References

Miller, D, SB Admin, viewed 25 April 2016,

http://startbootstrap.com/template-overviews/sb-admin/

Yong, M, K, Java SHA Hashing Example, viewed 25 April 2016,

http://www.mkyong.com/java/java-sha-hashing-example/