Requirements Engineering

Premier League System

Submitted By: Darren Godley (T00187846)

Computing with Multi Media

Date Submitted: 12/12/2017

**Table of Contents**

[1. Introduction/overview 4](#_Toc505241740)

[2. Functional Components 5](#_Toc505241741)

[3. User Requirements 6](#_Toc505241742)

[3.1. PremSYS will manage teams in the league 6](#_Toc505241743)

[3.2. PremSYS will manage league fixtures 6](#_Toc505241744)

[3.3. PremSYS will store teams details 6](#_Toc505241745)

[4. System Requirements 7](#_Toc505241746)

[4.1. System Level Use Case Diagram 7](#_Toc505241747)

[4.2. Manage Teams 7](#_Toc505241748)

[4.2.1. Register team 7](#_Toc505241749)

[4.2.2. Update Team 9](#_Toc505241750)

[4.2.3. Relegate Team 11](#_Toc505241751)

[4.3. Manage Fixtures 12](#_Toc505241752)

[4.3.1. Generate Fixtures 12](#_Toc505241753)

[4.3.2. Schedule Fixtures 14](#_Toc505241754)

[4.3.3. Enter Results 15](#_Toc505241755)

[4.3.4. Show League Table 17](#_Toc505241756)

[4.4. Show Teams 18](#_Toc505241757)

[4.4.1. Generate Team Profile 18](#_Toc505241758)

[5. System Model 20](#_Toc505241759)

[5.1. DFD Elements 20](#_Toc505241760)

[5.2. Level-0 DFD 21](#_Toc505241761)

[5.3. Level-1 DFD 22](#_Toc505241762)

[5.4. Level-2 DFD (Process P1: Manage Teams) 23](#_Toc505241763)

[5.5. Level-2 DFD (Process P2: Schedule Fixtures) 24](#_Toc505241764)

[5.6. Level-2 DFD (Process P3: Do Admin) 25](#_Toc505241765)

[6. Data Model (Class Diagram) 26](#_Toc505241766)

[6.1. Class Diagram 26](#_Toc505241767)

[6.2. Relational Schema 27](#_Toc505241768)

[6.3. Database Schema 28](#_Toc505241769)

[7. Conclusion 29](#_Toc505241770)

[8. Appendices 30](#_Toc505241771)

[8.1. Appendix A – Title 30](#_Toc505241772)

[8.2. Appendix B – Title 30](#_Toc505241773)

# Introduction/overview

PremSYS is a system that generates 20 teams fixtures in a league, it allows the user to schedule fixtures, enter results and play out a league season. Which the finish the bottom 3 teams in the league are relegated.

# Functional Components

The functional components for PremSYS are shown below:

# User Requirements

## PremSYS will manage teams in the league

* + 1. PremSYS will register a team in the league
    2. PremSYS will update a team’s details
    3. PremSYS will relegate a team from the league

## PremSYS will manage league fixtures

* + 1. PremSYS will generate fixtures
    2. PremSYS will schedule fixtures
    3. PremSYS will record fixture results
    4. PremSYS will show league table

## PremSYS will store teams details

* + 1. PremSYS will generate a team profile

# System Requirements

Include a brief overview of the system requirements.

Your hierarchy chart / User requirements summarise these……..

## System Level Use Case Diagram

A brief introduction of the high-level modules/components of the system…..

System level UC diagram to be included here…

## Manage Teams

This module provides functions to register, update and relegate a team in the Premier League.

### Register team

This function will register a team in the league

Admin

Manager

Includes

Extends

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Register Team** | |
| **Use Case Id** | 1.1 | |
| **Priority** | High | |
| **Source** | User | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** | Team Manager | |
| **Description** | This function registers a team in the league | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** User starts enter teams function  **Step 4:** User enters each teams details:   * Team\_Name * Home\_Ground * Manager * Email * Phone | **Step 2:** System determines the Team\_ID. If Team\_ID is < 21 proceed.  **Step 3:** System displays UI  **Step 5:** System validates team details:   * All fields must entered * Email address must have valid format * Team name must not already be registered   **Step 6:** Set each teams Points to zero  **Step 7:** Save team details in Team File:   * Team\_ID * Team\_Name * Home\_Ground * Manager * Email * Phone * Points   **Step 8:** Display confirmation message  **Step 9:** Clear the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Max Teams Entered** |  | **Step 2:** Team\_ID is 21  **Step 3:** Display message “Already 20 teams registered”  **Step 4:** Terminate function |
| **Field not entered** |  | **Step 4:** A blank field detected  **Step 5:** Display message “This field must be entered”  **Step 6:** Position cursor in offending field and return to step 4 |
| **Invalid Email** |  | **Step 4:** Invalid email address entered  **Step 5:** Display message “Enter a valid email address”  **Step 6:** Position cursor in offending field and return to step 4 |
| **Conclusions** | Teams are registered on the system | |
| **Post conditions** | Fixtures can now be generated | |
| **Business Rules** | Cannot be more than 20 teams | |
| **Implementation Constraints** |  | |

### Update Team

This function will update a team’s details in the league

Admin

Manager

Includes

Extends

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Update Team** | |
| **Use Case Id** | 1.2 | |
| **Priority** | High | |
| **Source** | User | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** | Team Manager | |
| **Description** | This function updates a team in the league | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** User starts update teams function  **Step 3:** User selects the team to be updated from the team file  **Step 4:** User enters the selected teams details:   * Team\_Name * Home\_Ground * Manager * Email * Phone | **Step 2:** System displays the teams  **Step 5:** System validates team details:   * All fields must entered * Email address must have valid format   **Step 6:** Update team details in Team File:   * Team\_Name * Home\_Ground * Manager * Email * Phone * Points   **Step 7:** Display confirmation message  **Step 8:** Clear the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 6:** A blank field detected  **Step 7:** Display message “This field must be entered”  **Step 8:** Position cursor in offending field and return to step 4 |
| **Invalid Email** |  | **Step 6:** Invalid email address entered  **Step 7:** Display message “Enter a valid email address”  **Step 8:** Position cursor in offending field and return to step 4 |
| **Conclusions** | The selected team is updated | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Relegate Team

This function will relegate a team from the league

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Relegate Team** | |
| **Use Case Id** | 1.3 | |
| **Priority** | High | |
| **Source** | User | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function relegates a team from the league | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** User starts relegate teams function  **Step 4:** User confirms relegation to be done | **Step 2:** System displays the teams   * Checks if all fixtures have been played. If true then moves to next step, else displays error message and closes the function.   **Step 3:** The system selects the three teams in the league with the lowest points from the team file and displays on UI  **Step 5:** Set all teams in the team files points to zero  **Step 6:** Display team that has been relegated  **Step 7:** Clear the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | A team is relegated | |
| **Post conditions** |  | |
| **Business Rules** | Can only be done at the end of the season after all fixtures have been played | |
| **Implementation Constraints** |  | |

## Manage Fixtures

This module provides functions to register and schedule fixtures. Enter results and show the Premier league table.

### Generate Fixtures

This function will generate fixtures in the league

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Generate Fixtures** | |
| **Use Case Id** | 2.1 | |
| **Priority** | High | |
| **Source** | User | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function generates the league fixtures | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** User starts generate fixtures function  **Step 5:** User confirm fixtures are to be generated | **Step 2:** System displays the UI  **Step 3:** System checks team file to confirm 20 teams are registered.  **Step 4:** System checks that all teams have zero points  **Step 6:** For each team in the Team File:   * Get first/next Team ID for opponents * Save fixtures in the fixtures file:   + Fixture\_ID   + Home\_Team\_ID   + Away\_Team\_ID   + Fixture\_Date (NULL)   + Fixture\_Time(NULL)   + Home\_Score (0)   + Away\_Score (0)   **Step 7:** Display confirmation message  **Step 8:** Clear the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 4:** A blank field detected  **Step 5:** Display message “This field must be entered”  **Step 6:** Position cursor in offending field and return to step 3 |
| **Team is playing themselves** |  | **Step 4:** Team\_ID is playing same Team\_ID  **Step 7:** Display message “Team can’t play themselves  **Step 8:** Position cursor in offending field and return to step 4 |
| **Conclusions** | The Fixture is generated | |
| **Post conditions** |  | |
| **Business Rules** | Must be 20 teams registered  All teams must have zero points | |
| **Implementation Constraints** |  | |

### Schedule Fixtures

This function will schedule fixtures in the league

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Schedule Fixtures** | |
| **Use Case Id** | 2.2 | |
| **Priority** | High | |
| **Source** | User | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function Schedules league fixtures | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** User starts schedulefixtures function  **Step 4:** User selects team for fixture to schedule  **Step 6:** User selects the fixture to schedule and enters:   * Fixture Date * Fixture Time | **Step 2:** System retrieves team details from Team File and loads on UI  **Step 3:** System displays the UI  **Step 5:** System retrieves details of all un-played fixtures for the selected team from the Fixtures File and Team File and displays on UI  **Step 7:** System validates data entered:   * Both fields must be entered * Fixture date must be between Today and 30 may of current season * Fixture time must be between 12:00pm and 8:00pm   **Step 8:** Update fixture in the fixtures file:   * Fixture Date * Fixture Time   **Step 9:** Display confirmation message  **Step 10:** Clear the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 4:** A blank field detected  **Step 5:** Display message “This field must be entered”  **Step 6:** Position cursor in offending field and return to step 3 |
| **Team is playing themselves** |  | **Step 4:** Team\_ID is playing same Team\_ID  **Step 7:** Display message “Team can’t play themselves  **Step 8:** Position cursor in offending field and return to step 4 |
| **Conclusions** | The Fixture is generated | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Enter Results

This function will record results in the league

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Enter Results** | |
| **Use Case Id** | 2.3 | |
| **Priority** | High | |
| **Source** | User | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function Enters the results | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** User starts enter results function  **Step 4:** User selects the fixture and enters:   * Home\_Score * Away\_Score | **Step 2:** System retrieves team details from Team File and loads on UI  **Step 3:** System displays the UI  **Step 5:** System validates data entered:   * Both fields must be entered   **Step 6:** Update result in the results file:   * Home\_Score * Away\_Score   **Step 7:** Display confirmation message  **Step 8:** Clear the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **Field not entered** |  | **Step 4:** A blank field detected  **Step 5:** Display message “This field must be entered”  **Step 6:** Position cursor in offending field and return to step 3 |
|  |  |  |
| **Conclusions** | Result is recorded | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### Show League Table

This function will show the league table

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Show League Table** | |
| **Use Case Id** | 2.4 | |
| **Priority** | High | |
| **Source** | User | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function displays the league table | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** User starts show league table function | **Step 2:** System displays the league table |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
|  |  |  |
| **Conclusions** | The league table is displayed | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Show Teams

### Generate Team Profile

This function will show the teams profile

Admin

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Generate Team Profile** | |
| **Use Case Id** | 3.1 | |
| **Priority** | High | |
| **Source** | User | |
| **Primary Business Actor** | Admin | |
| **Other Participating Actors** |  | |
| **Description** | This function displays the teams profile | |
| **Preconditions** |  | |
| **Trigger** | None | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** User starts the generate team profile function  **Step 4:** User selects the team | **Step 2:** System displays the teams in the league  **Step 3:** System asks the user to select which team’s profile he wants to display  **Step 5:** System displays team profile   * Team\_ID * Team\_Name * Home\_Ground * Manager * Email * Phone * Points   **Step 6:** Clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| **User doesn’t select any team** |  | **Step 5:** A blank field detected  **Step 6:** Display message “User must select a team”  **Step 6:** Position cursor in offending field and return to step 4 |
|  |  |  |
| **Conclusions** | The teams profile is displayed | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

# System Model

The following dataflow diagrams have been produced for the system.

## DFD Elements

External Entities

Teams

Manager

Data Stores

D1 Team File

D2 Fixture File

Processes

P1 Manage Teams

P1.1. Add Team

P1.2.Update Team

P1.3.Relegate Team

P2 Manage Fixtures

P2.1.Generate Fixtures

P2.2.Schedule Fixtures

P2.3.Enter Results

P2.4.Show League Table

P3 Do Admin

P3.1.Generate Team File

## Level-0 DFD

Team Details

Manager

Team Profile

PremSYS

## Level-1 DFD

Team Details

Match Results

Team Details

D1

Team File

D2

Fixtures File

Team Manager

Fixtures

P1

Teams

Team Details

Team Details

P2

Fixtures

P3

Admin

## Level-2 DFD (Process P1: Manage Teams)

Team Details

Team Details

Team Details

P1.3

Relegate Teams

Updated Team Details

Team Details

P1.2

Update Teams

Team Details

D1

Team File

P1.1

Add Teams

Team Manager

## Level-2 DFD (Process P2: Schedule Fixtures)

Team Details

P2.1

Generate Fixtures

Result

Fixture Details

Result

P2.4

Show League Table

Results

P2.3

Enter Results

Team Details

Fixtures

D2

Fixtures File

P2.2

Schedule Fixtures

D1

Team File

## Level-2 DFD (Process P3: Do Admin)

Points

Team Manager

Team Details

D1

Team File

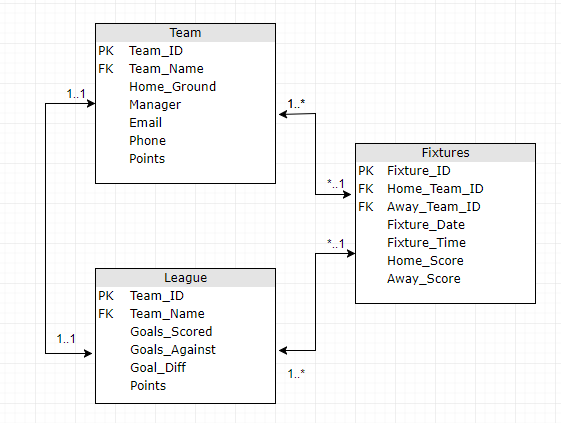
P3.1

Generate Team Profile

# Data Model (Class Diagram)

This is the data model for PremSYS

## Class Diagram



## Relational Schema

**Teams** (Team\_ID, Team\_Name, Home\_Ground, Manager, Email, Phone, Points)

**Fixtures** (Fixture\_ID, Home\_Team\_ID, Away\_Team\_ID, Fixture\_Date, Home\_Score, Away\_Score)

## Database Schema

**Schema:** PremSYS

**Relation: Team**

Attributes:

Team\_ID int (2) NOT NULL AUTO\_INCREMENT

Team\_Name char (20) NOT NULL

Home\_Ground char (20) NOT NULL

Manager char (15) NOT NULL

Email char (35) NOT NULL

Phone char (14) NOT NULL

Points int (2)

Primary Key Team\_ID

**Relation: Fixtures**

Attributes:

Fixture\_ID int (2) NOT NULL AUTO\_INCREMENT

Home\_Team\_ID int (2) NOT NULL

Away\_Team\_ID int (2) NOT NULL

Fixture\_Date date NOT NULL

Fixture\_Time char (5)

Home\_Score int (1)

Away\_Score int (1)

Primary Key Fixture \_ID

Foreign Key Home\_Team\_ID, Away\_Team\_ID References Teams

# Conclusion

# Appendices

## Appendix A – Title

## Appendix B – Title

Might include:

* **Lookup / Reference tables**
* **Sample reports / Listings**