

# Final Project Report

# Premier League Football Management System

[Group 24]

- R.G.S.Udara (PS/2019/027)
- Jadurshaa Subramaniam (PS/2019/076)
- Dilushika Anton Michael (PS/2019/090)
- Sachini Navodya Wannigama (PS/2019/177)
- A.M.A.D. Attanayake (PS/2019/201)

## Introduction

This Project is about a Premier League Football Management.

This is a software package which is helpful in the areas of Football.

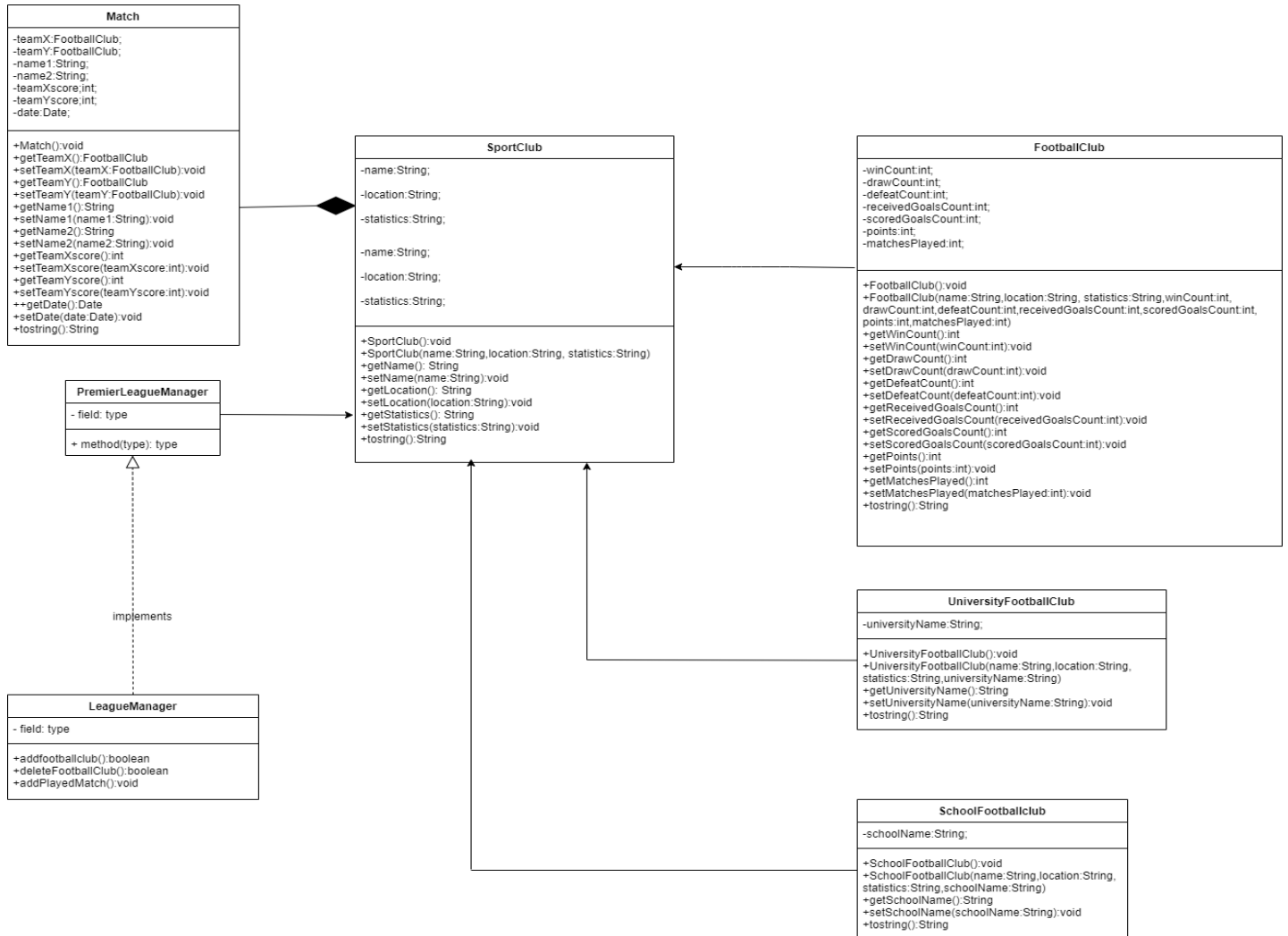
Maintaining the records of the players, coach, teams, points, matches etc. on paper is very difficult and time consuming. So, it is necessary to have a computerized system that manages all these issues.

Thus, basis of this project is to work on the football management system. Our main aim is to maintain records, generate reports and make analysis based on reports generated. This system is built to reduce complexity of system for the users handling the system.

Admin can login as administrator. This will allow admin to get full control to the System.

This software enables the admin to create or delete club, play a match, display statics for club, display premier league table, display GUI, randomly generate match, search matches from date played.

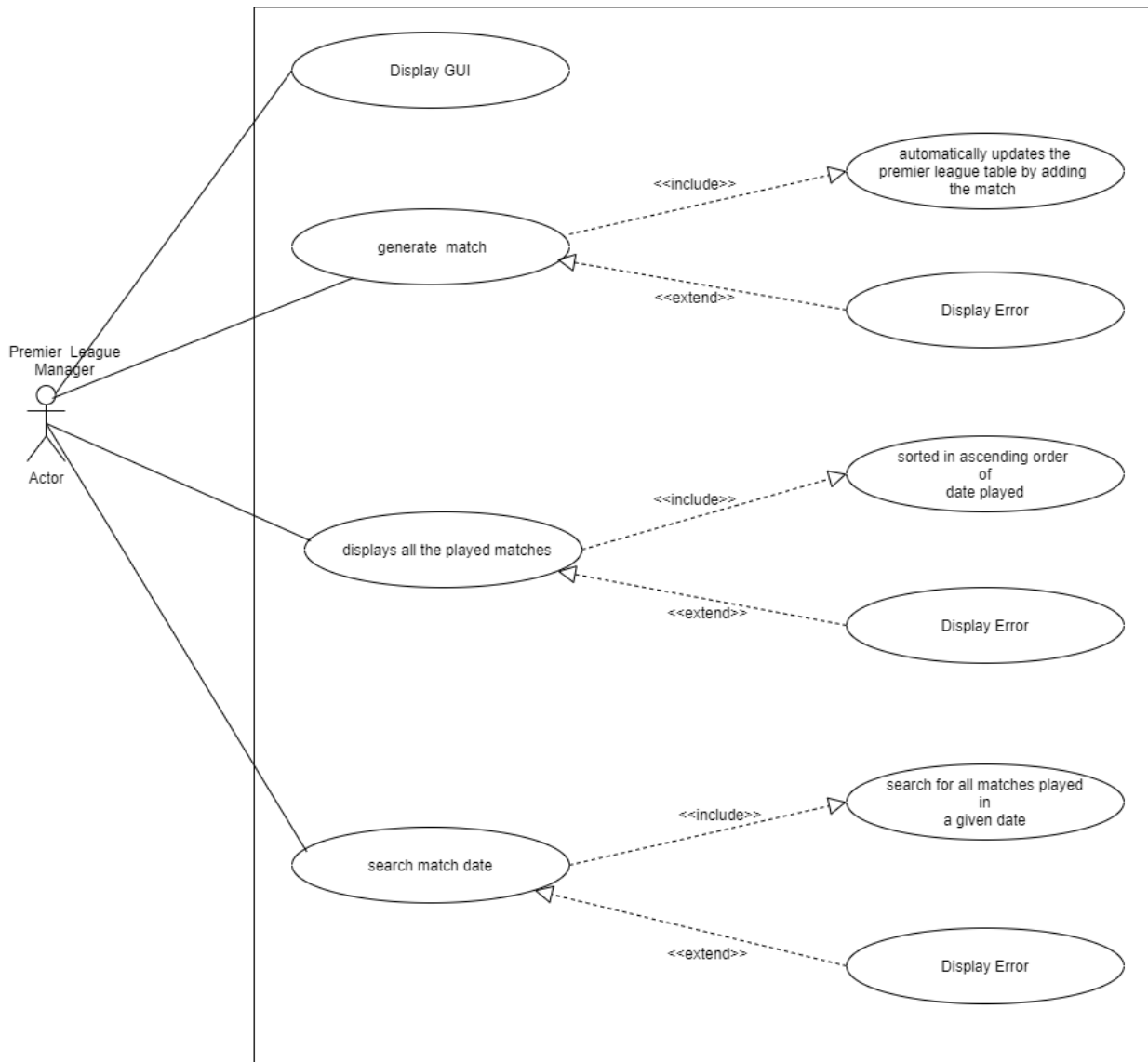
# Class Diagram



## Use case diagram for premier league manager



## Ues case diagram for GUI



## Java code

### SportClub.java

```
import java.io.Serializable;

import java.util.Objects;

abstract class SportsClub implements Serializable {

    private String name;

    private String location;

    private String statistics;

    public SportsClub(String name,String location,String statistics) {

        this.name = name;

        this.location=location;

        this.statistics=statistics;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public String getLocation() {

        return location;

    }

    public void setLocation(String location) {
```

```
        this.location = location;
    }
}
```

```
public String getStatistics() {
    return statistics;
}
```

```
public void setStatistics(String statistics) {
    this.statistics = statistics;
}
```

```
@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    SportsClub that = (SportsClub) o;
    return Objects.equals(name, that.name) &&
        Objects.equals(location, that.location) &&
        Objects.equals(statistics, that.statistics);
}
```

```
@Override
public int hashCode() {
    return Objects.hash(name, location, statistics);
}
```

```
@Override
```



```

public String toString() {
    return "SportsClub{" +
        "name=" + name + "\" +
        ", location=" + location + "\" +
        ", statistics=" + statistics + "\" +
        '}'";
}
}

```

### FootballClub.java

```

import java.io.Serializable;
import java.util.Objects;

public class FootballClub extends SportsClub implements Serializable {

    private int winCount;
    private int drawCount;
    private int defeatCount;
    private int receivedGoalsCount;
    private int scoredGoalsCount;
    private int points;
    private int matchesPlayed;

    public FootballClub(String name,String location) {
        super(name,location,"");
        this.winCount=0;
        this.drawCount=0;
        this.defeatCount=0;
        this.receivedGoalsCount=0;
    }
}

```

```
    this.scoredGoalsCount=0;

    this.points=0;

    this.matchesPlayed=0;

}
```

```
    public FootballClub(String name, String location, String statistics, int winCount,int drawCount,int
defeatCount,int receivedGoalsCount,int scoredGoalsCount,int points,int matchesPlayed) {

        super(name, location, statistics);

        this.winCount = winCount;

        this.drawCount=drawCount;

        this.defeatCount=defeatCount;

        this.receivedGoalsCount=receivedGoalsCount;

        this.scoredGoalsCount=scoredGoalsCount;

        this.points=points;

        this.matchesPlayed=matchesPlayed;

    }
```

```
    public int getWinCount() {

        return winCount;

    }
```

```
    public void setWinCount(int winCount) {

        this.winCount = winCount;

    }
```

```
    public int getDrawCount() {

        return drawCount;
```

```
}
```

```
public void setDrawCount(int drawCount) {  
    this.drawCount = drawCount;  
}
```

```
public int getDefeatCount() {  
    return defeatCount;  
}
```

```
public void setDefeatCount(int defeatCount) {  
    this.defeatCount = defeatCount;  
}
```

```
public int getReceivedGoalsCount() {  
    return receivedGoalsCount;  
}
```

```
public void setReceivedGoalsCount(int receivedGoalsCount) {  
    this.receivedGoalsCount = receivedGoalsCount;  
}
```

```
public int getScoredGoalsCount() {  
    return scoredGoalsCount;  
}
```

```
public void setScoredGoalsCount(int scoredGoalsCount) {  
    this.scoredGoalsCount = scoredGoalsCount;  
}
```

```
}
```

```
public int getPoints() {  
    return points;  
}
```

```
public void setPoints(int points) {  
    this.points = points;  
}
```

```
public int getMatchesPlayed() {  
    return matchesPlayed;  
}
```

```
public void setMatchesPlayed(int matchesPlayed) {  
    this.matchesPlayed = matchesPlayed;  
}
```

@Override

```
public boolean equals(Object o) {  
    if (this == o) return true;  
    if (o == null || getClass() != o.getClass()) return false;  
    FootballClub that = (FootballClub) o;  
    return winCount == that.winCount &&  
        drawCount == that.drawCount &&  
        defeatCount == that.defeatCount &&  
        receivedGoalsCount == that.receivedGoalsCount &&  
        scoredGoalsCount == that.scoredGoalsCount &&  
        points == that.points &&
```

```

        matchesPlayed == that.matchesPlayed;
    }

    @Override
    public int hashCode() {
        return Objects.hash(winCount, drawCount, defeatCount, receivedGoalsCount, scoredGoalsCount,
points, matchesPlayed);
    }

    @Override
    public String toString() {
        return "FootballClub{" +
            "winCount=" + winCount +
            ", drawCount=" + drawCount +
            ", defeatCount=" + defeatCount +
            ", receivedGoalsCount=" + receivedGoalsCount +

            " , scoredGoalsCount=" + scoredGoalsCount +
            " , points=" + points +
            " , matchesPlayed=" + matchesPlayed +
            '}';
    }
}

```

#### UniversityFootballClub.java

```

import java.util.Objects;

public class UniversityFootballClub extends SportsClub {
    private String universityName;

```

```

public UniversityFootballClub(String universityName,String name,String location,String statistics) {
    super(name,location,statistics);
    this.universityName = universityName;
}

public String getUniversityName() {
    return universityName;
}

public void setUniversityName(String universityName) {
    this.universityName = universityName;
}

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    UniversityFootballClub that = (UniversityFootballClub) o;
    return Objects.equals(universityName, that.universityName);
}

@Override
public int hashCode() {
    return Objects.hash(universityName);
}

@Override
public String toString() {
    return "UniversityFootballClub{" +
        "universityName=\"" + universityName + "\"" +
        '}';
}
}

```

### SchoolFootballClub.java

```

import java.util.Objects;

public class SchoolFootballClub extends SportsClub{
    private String schoolName;

    public SchoolFootballClub(String schoolName,String name,String location,String statistics) {
        super(name,location,statistics);
    }
}

```

```

        this.schoolName = schoolName;
    }

    public String getSchoolName() {
        return schoolName;
    }

    public void setSchoolName(String schoolName) {
        this.schoolName = schoolName;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        SchoolFootballClub that = (SchoolFootballClub) o;
        return Objects.equals(schoolName, that.schoolName);
    }

    @Override
    public int hashCode() {
        return Objects.hash(schoolName);
    }

```

```

@Override
public String toString() {
    return "SchoolFootballClub{" +
        "schoolName='" + schoolName + "\" +
        '}'";
}
}

```

### Match.java

```

import java.io.Serializable;
import java.util.Date;
import java.util.Objects;

public class Match implements Serializable {
    private FootballClub teamX;
    private FootballClub teamY;
    private String name1;
    private String name2;
    private int teamXScore;

```

```

private int teamYScore;
private Date date;

public Match() {

}

public FootballClub getTeamX() {
    return teamX;
}

public void setTeamX(FootballClub teamX) {
    this.teamX = teamX;
}

public FootballClub getTeamY() {
    return teamY;
}

public void setTeamY(FootballClub teamY) {
    this.teamY = teamY;
}

public int getTeamXScore() {
    return teamXScore;
}

public void setTeamXScore(int teamXScore) {
    this.teamXScore = teamXScore;
}

public int getTeamYScore() {
    return teamYScore;
}

public void setTeamYScore(int teamYScore) {
    this.teamYScore = teamYScore;
}

public Date getDate() {
    return date;
}

public void setDate(Date date) {
    this.date = date;
}

public String getName1() {

```



```

        return name1;
    }

    public String getName2() {
        return name2;
    }

    public void setName2(String name2) {
        this.name2 = name2;
    }

    public void setName1(String name1) {
        this.name1 = name1;
    }

    @Override
    public String toString() {
        return "Match{" +
            "teamX=" + teamX +
            ", teamY=" + teamY +
            ", xName=\"" + name1 + "\" +
            ", yName=\"" + name2 + "\" +
            ", teamXScore=" + teamXScore +
            ", teamYScore=" + teamYScore +
            ", date=" + date +
            '}';
    }
}

```

### CusComparator.java

```

import java.util.Comparator;

public class CusComparator implements Comparator<FootballClub> {

    @Override
    public int compare(FootballClub f1, FootballClub f2) {

        if (f1.getPoints()>f2.getPoints())
            return -1;
        else if (f1.getPoints()<f2.getPoints())
            return 1;
        else {
            int goaldiffer1=f1.getScoredGoalsCount()-f1.getReceivedGoalsCount();
            int goaldiffer2=f2.getScoredGoalsCount()-f2.getReceivedGoalsCount();

```

```

        if (goaldiffer1>goaldiffer2)
            return -1;
        else if (goaldiffer1<goaldiffer2)
            return 1;
        else return 0;
    }
}
}

```

### LeagueManager.java

```

import java.util.Date;
import java.util.List;

public interface LeagueManager {
    public boolean addfootballclub(FootballClub club);
    public boolean deleteFootballClub(String line);
    public void AddPlayedMatch(Match match);
    public void RunGUI();

    Match getMatchbyDate(Date date);
}

```

### PremierLeagueManager.java

```

import java.io.*;
import java.util.*;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.List;

public class PremierLeagueManager implements LeagueManager, Serializable {

    private static final long serialVersionUID = 1L;
    // private List<SportsClub> clubList = new ArrayList<SportsClub>();
    // private List<Match>clubList1=new ArrayList<>();

    public final ArrayList<FootballClub> league;
    public final ArrayList<Match> matchArrayList;
    private final Scanner scanner = null;
    private FootballClub[] getclubList;
    private Match[] getclubList1;
}

```

```
public PremierLeagueManager() {  
    this.matchArrayList = new ArrayList<Match>();  
    league = new ArrayList<FootballClub>();  
}
```

```
@Override  
public boolean addfootballclub(FootballClub club) {  
  
    for (FootballClub club1 : league) {  
        if (Objects.equals(club1.getName(), club.getName())) {  
  
            return false;  
        }  
    }  
    league.add(club);  
    return true;  
}
```

```
@Override  
public boolean deleteFootballClub(String clubname) {  
  
    for (FootballClub club1 : league) {  
  
        if (Objects.equals(club1.getName(), clubname)) {  
            league.remove(club1);  
            return true;  
        }  
    }  
    return false;  
}
```

```
@Override  
public Match getMatchbyDate(Date date){  
  
    for (Match match : matchArrayList) {  
  
        if (Objects.equals(match.getDate(), date)) {  
  
            return match;  
        }  
    }  
}
```

```

    }

    return null;
}

@Override
public void AddPlayedMatch(Match match) {
    this.matchArrayList.add(match);
}

@Override
public void RunGUI() {

}
}

```

## Main.java

```

import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

import java.io.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.*;

public class Main extends Application {
    Stage window;
    TableView<FootballClub> tableView;
    PremierLeagueManager plm= null;
}

```

```

Scanner scanner;

public static void main(String[] args) {

    launch();

}

@Override
public void start(Stage primaryStage) throws Exception {

    File filename = new File("./premierLeague.ser");
    if (filename.exists()) {
        try {
            // Reading the object from a file
            FileInputStream file = new FileInputStream(filename);
            ObjectInputStream in = new ObjectInputStream(file);

            // Method for deserialization of object
            plm = (PremierLeagueManager) in.readObject();
            display();
            in.close();
            file.close();

            System.out.println("Object has been deserialized ");
//            System.out.println("size = " + premierLeagueManager.clubList.size());

        } catch (IOException ex) {
            System.out.println("IOException is caught");
        } catch (ClassNotFoundException ex) {
            System.out.println("ClassNotFoundException is caught");
        }
    }

    } else {

        plm = new PremierLeagueManager();
        display();
    }
}

public boolean display() {
    scanner = new Scanner(System.in);
    while (true) {
        System.out.println("Press 1: Create a new football club and add it to premier league ");
        System.out.println("Press 2: Delete an existing club from premier league ");
    }
}

```

```

System.out.println("Press 3: Display statistics for club ");
System.out.println("Press 4: Display Premier League Table ");
System.out.println("Press 5: Add Played A Match");
System.out.println("Press 6: Run GUI");
String line = scanner.nextLine();
int command = 0;
try {
    command = Integer.parseInt(line);
} catch (Exception e) {

}

switch (command) {
    case 1:
        Addfootballclub();
        break;
    case 2:
        DeleteFootballClub();
        break;

    case 3:
        DisplayStatistics();
        break;
    case 4:
        DisplayPremierLeagueTable();
        break;
    case 5:
        AddPlayedMatch();
        break;
    case 6:
        RunGUI();
        break;

    default:
        System.out.println("Wrong command");
}

}
}

```

```
//add football club
```

```
private void Addfootballclub() {
```

```
    FootballClub club;
```

```

System.out.println("Enter your club name: ");
String name = scanner.nextLine();

System.out.println("Enter your club location");
String location = scanner.nextLine();

club = new FootballClub(name, location);

if (plm.addfootballclub(club)) {
    System.out.println("This club is successfully added to the league");
    System.out.println("-----");
} else {
    System.out.println("This club is already exist");
    System.out.println("-----");
}
saveFile();
}

//delete football club

private void DeleteFootballClub() {
    System.out.println("Enter your club name: ");
    String line = scanner.nextLine();

    if (plm.deleteFootballClub(line)) {
        System.out.println("Deleted Successfully");
    } else {
        System.out.println("Team is not exist");
    }
    saveFile();
}

//display statistics

private void DisplayStatistics() {

    System.out.println("Insert your club name: ");
    String line = scanner.nextLine();
    FootballClub ft = null;
    for (FootballClub footballClub : plm.league) {
        if (footballClub.getName().equals(line)) {
            ft = footballClub;
        }
    }
    if (ft != null) {
        System.out.println("FootballClub " + ft.getName() + " Won matches: " + ft.getWinCount());
    }
}

```

```

        System.out.println("FootballClub " + ft.getName() + " Draw matches: " + ft.getDrawCount());
        System.out.println("FootballClub " + ft.getName() + " Defeat matches: " + ft.getDefeatCount());
        System.out.println("FootballClub " + ft.getName() + " ReceivedGoals : " +
ft.getReceivedGoalsCount());
        System.out.println("FootballClub " + ft.getName() + " ScoredGoals : " + ft.getScoredGoalsCount());
        System.out.println("FootballClub " + ft.getName() + " Points : " + ft.getPoints());
        System.out.println("FootballClub " + ft.getName() + " MatchesPlayed : " + ft.getMatchesPlayed());
        System.out.println("-----");
    } else {
        System.out.println("No such club in league");
        System.out.println("-----");
    }
}
}

```

//display PremierLeagueTable

```
private void DisplayPremierLeagueTable() {
```

```

    Collections.sort(plm.league, new CusComparator());
    for (FootballClub footballClub : plm.league) {
        System.out.println("Club Name: " + footballClub.getName() + " Points: " + footballClub.getPoints()
+ " Goal Difference: " + (footballClub.getScoredGoalsCount() - footballClub.getReceivedGoalsCount()));
    }
    System.out.println("-----");
}

```

//display add played match

```
private void AddPlayedMatch() {
```

```

    System.out.println("Please Enter Date (format yyyy/mm/dd): ");
    String line = scanner.nextLine();
    Date date;
    try {
        date = new SimpleDateFormat("yyyy/mm/dd").parse(line);
    } catch (ParseException ex) {
        System.out.println("you have to enter date correct format");
        return;
    }
}

```



```

System.out.println("Please Enter Home Team Name: ");
line = scanner.nextLine();
FootballClub Home = null;
for (FootballClub footballClub : plm.league) {
    if (footballClub.getName().equals(line))
        Home = footballClub;
}
if (Home == null) {
    System.out.println("No such club in league");
    return;
}
System.out.println("Please Enter Away Team Name: ");
line = scanner.nextLine();
FootballClub Away = null;
for (FootballClub footballClub : plm.league) {
    if (footballClub.getName().equals(line))
        Away = footballClub;
}

if (Away == null) {
    System.out.println("No such club in league");
    return;
}
System.out.println("Enter Home team goals: ");
line = scanner.nextLine();
int HomeGoals = -1;
try {
    HomeGoals = Integer.parseInt(line);
} catch (Exception exception) {

}
if (HomeGoals == -1) {
    System.out.println("You have to enter number of home goals");
    return;
}
System.out.println("Enter Away team goals: ");
line = scanner.nextLine();
int AwayGoals = -1;
try {
    AwayGoals = Integer.parseInt(line);
} catch (Exception exception) {

}
if (AwayGoals == -1) {
    System.out.println("You have to enter number of away goals");
    return;
}

```

```

    }
    Match match = new Match();
    match.setName1(Home.getName());
    match.setName2(Away.getName());

    match.setDate(date);
    match.setTeamX(Home);
    match.setTeamY(Away);
    match.setTeamXScore(AwayGoals);
    match.setTeamYScore(HomeGoals);

    Home.setScoredGoalsCount(Home.getScoredGoalsCount() + HomeGoals);
    Away.setScoredGoalsCount(Away.getScoredGoalsCount() + AwayGoals);
    Home.setReceivedGoalsCount(Home.getReceivedGoalsCount() + AwayGoals);
    Away.setReceivedGoalsCount(Away.getReceivedGoalsCount() + HomeGoals);

    if (HomeGoals > AwayGoals) {
        Home.setPoints(Home.getPoints() + 3);
        Home.setWinCount(Home.getWinCount() + 1);
        Away.setDefeatCount(Away.getDefeatCount() + 1);
    } else if (HomeGoals < AwayGoals) {
        Away.setPoints(Away.getPoints() + 3);
        Away.setWinCount(Away.getWinCount() + 1);
        Home.setDefeatCount(Home.getDefeatCount() + 1);
    } else {
        Home.setPoints(Home.getPoints() + 1);
        Home.setDrawCount(Home.getDrawCount() + 1);
        Away.setPoints(Away.getPoints() + 1);
        Away.setDrawCount(Away.getDrawCount() + 1);
    }
    System.out.println("-----");
    plm.AddPlayedMatch(match);
    saveFile();
}

//save file

public void saveFile() {
    File filename = new File("./premierLeague.ser");

    try {
        //Saving of object in a file
        FileOutputStream file = new FileOutputStream(filename);

```

```

        ObjectOutputStream out = new ObjectOutputStream(file);
        // Method for serialization of object
        out.writeObject(plm);
        out.close();
        file.close();
        System.out.println("Object has been serialized");

    } catch (IOException ex) {
        System.out.println("IOException is caught");
    }
}

private void RunGUI() {

    //Creating a table view
    TableView<FootballClub> teamtable=new TableView<>();

    ObservableList<FootballClub> teams=FXCollections.observableArrayList();

    //Setting the size of the table
    TableView tableView = new TableView();
    tableView.setLayoutX(95);
    tableView.setLayoutY(100);
    tableView.setPrefSize(1000, 500);

    //Creating columns
    TableColumn<FootballClub, String> NameColum = new TableColumn<>("Name");
    NameColum.setMaxWidth(400);
    NameColum.setCellValueFactory(new PropertyValueFactory<>("name"));

    TableColumn<FootballClub, Integer> WinCountColum=new TableColumn<>("WinCount");
    WinCountColum.setMaxWidth(400);
    WinCountColum.setCellValueFactory(new PropertyValueFactory<>("WinCount"));
    WinCountColum.setSortType(TableColumn.SortType.DESENDING);

    TableColumn<FootballClub, Integer> DrawCountColum = new TableColumn<>("DrawCount");
    DrawCountColum.setMinWidth(100);
    DrawCountColum.setCellValueFactory(new PropertyValueFactory<>("drawCount"));

    TableColumn<FootballClub, Integer> DefeatCountColum = new TableColumn<>("DefeatCount");
    DefeatCountColum.setMinWidth(100);
    DefeatCountColum.setCellValueFactory(new PropertyValueFactory<>("defeatCount"));

```

```

        TableColumn<FootballClub, Integer> ReceivedGoalsCountColumn = new
TableColumn<>("ReceivedGoalsCount");
        ReceivedGoalsCountColumn.setMinWidth(180);
        ReceivedGoalsCountColumn.setCellValueFactory(new
PropertyConnectionFactory<>("receivedGoalsCount"));

        TableColumn<FootballClub, Integer> ScoredGoalsCountColumn = new
TableColumn<>("ScoredGoalsCount");
        ScoredGoalsCountColumn.setMinWidth(180);
        ScoredGoalsCountColumn.setCellValueFactory(new PropertyConnectionFactory<>("scoredGoalsCount"));

        TableColumn<FootballClub, Integer> PointsColumn = new TableColumn<>("Points");
        PointsColumn.setMinWidth(80);
        PointsColumn.setCellValueFactory(new PropertyConnectionFactory<>("points"));

        TableColumn<FootballClub, Integer> MatchesPlayedColumn = new
TableColumn<>("MatchesPlayed");
        MatchesPlayedColumn.setMinWidth(130);
        MatchesPlayedColumn.setCellValueFactory(new PropertyConnectionFactory<>("matchesPlayed"));


        Random random=new Random();

        Button addbutton = new Button("Generate Match");

        addbutton.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {

                generateRandomMatch();

            }
        });

        Button dbutton = new Button("Date Played ");

        HBox hBox10 = new HBox();
        hBox10.setPadding(new Insets(40, 100, 40, 100));
        hBox10.setSpacing(20);
        hBox10.getChildren().addAll( addbutton, dbutton);

        //tableView.setItems(getFootballClub());

        tableView.getColumns().add(NameColumn);

```

```

tableView.getColumns().add(WinCountColumn);
tableView.getColumns().add(DrawCountColumn);
tableView.getColumns().add(DefeatCountColumn);
tableView.getColumns().add(ReceivedGoalsCountColumn);
tableView.getColumns().add(ScoredGoalsCountColumn);
tableView.getColumns().add(PointsColumn);
tableView.getColumns().add(MatchesPlayedColumn);

```

```

VBox vbox = new VBox();
vbox.getChildren().addAll(tableView, hbox10);

```

```

for (FootballClub footballClub : plm.league) {
    tableView.getItems().add(footballClub);
}

```

```

Pane pane1 = new Pane();
pane1.setStyle("-fx-background-color:#20B2AA");

```

```

pane1.getChildren().add(tableView);
pane1.getChildren().add(vbox);
pane1.getChildren().add(hbox10);

```

```

Button backbtton=new Button("Back");
backbtton.setLayoutX(100);
backbtton.setLayoutY(50);

```

```

Label lb2=new Label("Search Date");
lb2.setLayoutX(200);
lb2.setLayoutY(52);

```

```

Button searchbtton=new Button("Search");
searchbtton.setLayoutX(600);
searchbtton.setLayoutY(50);

```

```

TextField tf1 = new TextField();
tf1.setPromptText("YYYY/MM/DD");
tf1.setLayoutX(350);
tf1.setLayoutY(50);

```

```

//Creating a table view
TableView<Match> match=new TableView<>();

```

```

ObservableList<Match> matches=FXCollections.observableArrayList();

//Setting the size of the table
TableView tableView1 = new TableView();
tableView1.setLayoutX(95);
tableView1.setLayoutY(100);
tableView1.setPrefSize(1000, 500);

//Creating columns
TableColumn<Match, String> TeamXColumn = new TableColumn<>("TeamX Name");
TeamXColumn.setMaxWidth(400);
TeamXColumn.setCellValueFactory(new PropertyValueFactory<>("name1"));

TableColumn<Match, String> TeamYColumn = new TableColumn<>("TeamY Name");
TeamYColumn.setMaxWidth(400);
TeamYColumn.setCellValueFactory(new PropertyValueFactory<>("name2"));

TableColumn<Match, Integer> TeamXScoreColumn=new TableColumn<>("TeamXScore");
TeamXScoreColumn.setMaxWidth(400);
TeamXScoreColumn.setCellValueFactory(new PropertyValueFactory<>("teamXScore"));
TeamXScoreColumn.setSortType(TableColumn.SortType.DESENDING);

TableColumn<Match, Integer> TeamYScoreColumn = new TableColumn<>("TeamYScore");
TeamYScoreColumn.setMinWidth(100);
TeamYScoreColumn.setCellValueFactory(new PropertyValueFactory<>("teamYScore"));

TableColumn<Match, Date> DateColumn = new TableColumn<>("Date");
DateColumn.setMinWidth(300);
DateColumn.setCellValueFactory(new PropertyValueFactory<>("date"));

tableView1.getColumns().add(TeamXColumn);
tableView1.getColumns().add(TeamYColumn);
tableView1.getColumns().add(TeamXScoreColumn);
tableView1.getColumns().add(TeamYScoreColumn);
tableView1.getColumns().add(DateColumn);

VBox vBox1 = new VBox();
vBox1.getChildren().addAll(tableView1);

for (Match match1 : plm.matchArrayList) {

    tableView1.getItems().add(match1);
}

```

```

searchbttton.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {

        tableView1.getItems().clear();
        Match found = null;
        String seaching = tf1.getText();
        try {
            found = plm.getMatchbyDate(new SimpleDateFormat("yyyy/mm/dd").parse(seaching));
        } catch (ParseException e) {
            System.out.println("Your Format is wrong");
        }
        if(found != null) {
            tableView1.getItems().add(found);

            Alert a = new Alert(Alert.AlertType.NONE);
            a.setAlertType(Alert.AlertType.CONFIRMATION);

            a.setContentText("random match generated successfully !!!\n");
            tf1.clear();
        } else {
            System.out.println("No search results");
        }

    }
});

Pane pane2=new Pane();
pane2.setStyle("-fx-background-color:#20B2AA");
pane2.getChildren().add(backbttton);
pane2.getChildren().add(searchbttton);
pane2.getChildren().add(tf1);
pane2.getChildren().add(lb2);
pane2.getChildren().add(tableView1);

BorderPane borderPane = new BorderPane();
borderPane.setCenter(pane1);

dbutton.setOnAction(event -> {
    borderPane.setCenter(pane2);
});

backbttton.setOnAction(event -> {
    borderPane.setCenter(pane1);

```

```

});

//Setting the scene
Stage stage = new Stage();
Scene scene = new Scene(borderPane, 1200, 800);
stage.setTitle("Football Management System");
stage.setScene(scene);
stage.showAndWait();

}

private void generateRandomMatch() {

    Random rm2=new Random();
    int rmitem1=rm2.nextInt(plm.league.size());

    FootballClub rmClub1=plm.league.get(rmitem1);

    System.out.println(""+ rmClub1.getName());
    boolean same = true;
    FootballClub rmClub2 = null;
    while(same){
        int rmitem2=rm2.nextInt(plm.league.size());
        if(rmitem2 == rmitem1){
            same= true;
        }else{
            same = false;
            rmClub2=plm.league.get(rmitem2);
            System.out.println(""+ rmClub2.getName());
        }
    }
    int scr1=rm2.nextInt(10);
    int scr2=rm2.nextInt(10);
    System.out.println(scr1);
    System.out.println(scr2);

    long sD = new Date("01/01/2020").getTime();
    long eD = new Date("01/01/2021").getTime();

    //generate a random long with this start and end date
    Random r = new Random();
    long random = eD + (long) (r.nextDouble() * (sD - eD));

    Date resultdate = new Date(random);

```



```

        System.out.println(new SimpleDateFormat("yyyy/MM/dd").format(resultdate));

        addRandomMatch(rmClub1,scr1,rmClub2,scr2,resultdate);

    }

    private void addRandomMatch(FootballClub Home,int HomeGoals,FootballClub Away,int
AwayGoals,Date date) {
        Match match = new Match();
        match.setName1(Home.getName());
        match.setName2(Away.getName());
        match.setDate(date);
        match.setTeamX(Home);
        match.setTeamY(Away);
        match.setTeamXScore(AwayGoals);
        match.setTeamYScore(HomeGoals);

        Home.setScoredGoalsCount(Home.getScoredGoalsCount() + HomeGoals);
        Away.setScoredGoalsCount(Away.getScoredGoalsCount() + AwayGoals);
        Home.setReceivedGoalsCount(Home.getReceivedGoalsCount() + AwayGoals);
        Away.setReceivedGoalsCount(Away.getReceivedGoalsCount() + HomeGoals);

        if (HomeGoals > AwayGoals) {
            Home.setPoints(Home.getPoints() + 3);
            Home.setWinCount(Home.getWinCount() + 1);
            Away.setDefeatCount(Away.getDefeatCount() + 1);
        } else if (HomeGoals < AwayGoals) {
            Away.setPoints(Away.getPoints() + 3);
            Away.setWinCount(Away.getWinCount() + 1);
            Home.setDefeatCount(Home.getDefeatCount() + 1);
        } else {
            Home.setPoints(Home.getPoints() + 1);
            Home.setDrawCount(Home.getDrawCount() + 1);
            Away.setPoints(Away.getPoints() + 1);
            Away.setDrawCount(Away.getDrawCount() + 1);
        }
        plm.AddPlayedMatch(match);
        System.out.println("-----");
        saveFile();

    }

    public ObservableList<FootballClub> getFootballClub() {

```

```

        ObservableList<FootballClub> footballClubs = FXCollections.observableArrayList();

        return footballClubs;
    }

    public ObservableList<Match> getMatch(){
        ObservableList<Match> matches=FXCollections.observableArrayList();
        return matches;
    }
}

```

#### PremierLeagueManagerTest.java

```

import java.text.ParseException;
import java.text.SimpleDateFormat;

import static org.junit.jupiter.api.Assertions.*;

class PremierLeagueManagerTest {
    PremierLeagueManager plm;

    @org.junit.jupiter.api.BeforeEach
    void setUp() {
        plm = new PremierLeagueManager();
    }

    @org.junit.jupiter.api.Test
    void addfootballclub() {
        FootballClub ft1 = new FootballClub("qq", "yy");
        plm.addfootballclub(ft1);

        assertEquals(1, plm.league.size());
    }

    @org.junit.jupiter.api.Test
    void addExistingfootballclub() {
        FootballClub ft1 = new FootballClub("qq", "yy");
        plm.addfootballclub(ft1);
    }
}

```

```

FootballClub ft2 = new FootballClub("qq", "yy");

boolean sucess= plm.addfootballclub(ft2);

assertEquals(false,sucess);
}

@org.junit.jupiter.api.Test
void deleteFootballClub() {
    FootballClub ft1 = new FootballClub("qq", "yy");
    plm.addfootballclub(ft1);

    plm.deleteFootballClub(ft1.getName());

    assertEquals(0, plm.league.size());
}

@org.junit.jupiter.api.Test
void deleteNotExistingFootballClub() {
    FootballClub ft1 = new FootballClub("qq", "yy");
    plm.addfootballclub(ft1);

    boolean sucess=plm.deleteFootballClub("ww");

    assertEquals(false,sucess);
}

@org.junit.jupiter.api.Test
void AddPlayedMatch() {
    Match match = new Match();
    plm.AddPlayedMatch(match);

    assertEquals(1, plm.matchArrayList.size());
}

@org.junit.jupiter.api.Test
void getMatchbyDateExisting() throws ParseException {

    Match match = new Match();
    match.setDate(new SimpleDateFormat("yyyy/mm/dd").parse("2020/12/20"));
    plm.AddPlayedMatch(match);
}

```

```

        Match search = plm.getMatchbyDate(new
SimpleDateFormat("yyyy/mm/dd").parse("2020/12/20"));

        assertNotNull(search);
    }

    @org.junit.jupiter.api.Test
    void getMatchbyDateNotExisting() throws ParseException {

        Match match = new Match();
        match.setDate(new SimpleDateFormat("yyyy/mm/dd").parse("2020/12/20"));
        plm.AddPlayedMatch(match);

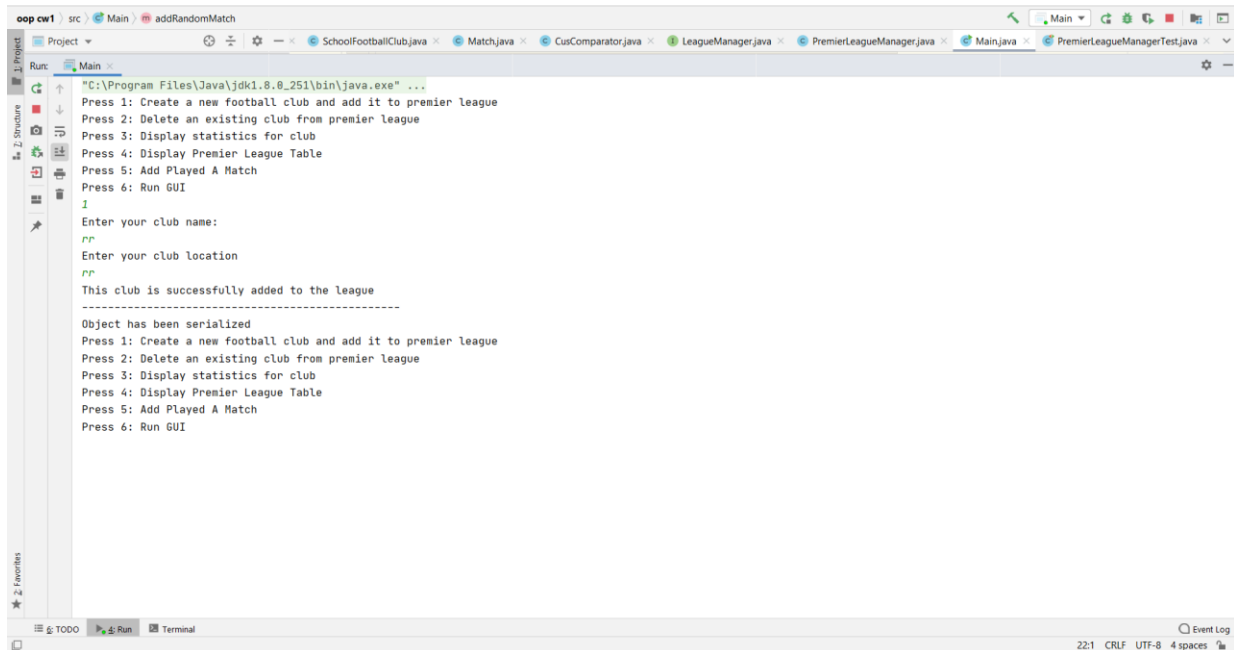
        Match search = plm.getMatchbyDate(new
SimpleDateFormat("yyyy/mm/dd").parse("2020/12/21"));

        assertNull(search);
    }
}

```

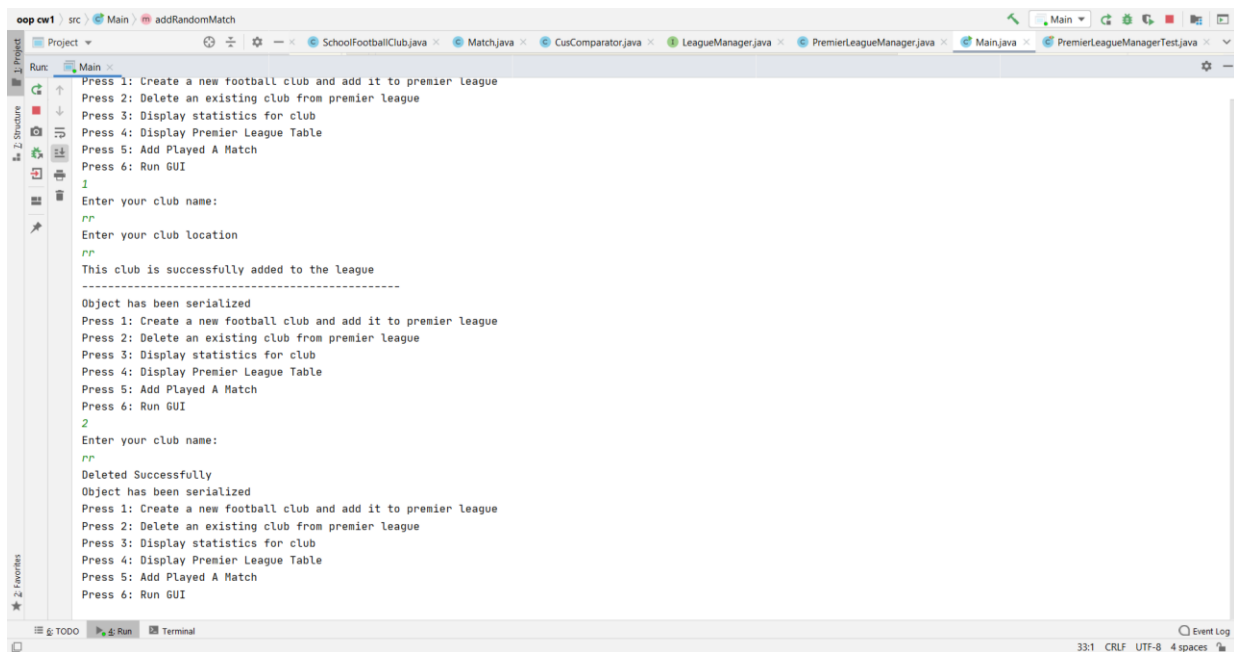
## 5.PremierLeagueManager Outputs

## 5.1 Add a Club



```
oop cw1 | src | Main | addRandomMatch
Project | Main | SchoolFootballClub.java | Match.java | CusComparator.java | LeagueManager.java | PremierLeagueManager.java | Main.java | PremierLeagueManagerTest.java
Run: Main
"C:\Program Files\Java\jdk1.8.0_251\bin\java.exe" ...
Press 1: Create a new football club and add it to premier league
Press 2: Delete an existing club from premier league
Press 3: Display statistics for club
Press 4: Display Premier League Table
Press 5: Add Played A Match
Press 6: Run GUI
1
Enter your club name:
PP
Enter your club location
PP
This club is successfully added to the league
-----
Object has been serialized
Press 1: Create a new football club and add it to premier league
Press 2: Delete an existing club from premier league
Press 3: Display statistics for club
Press 4: Display Premier League Table
Press 5: Add Played A Match
Press 6: Run GUI
```

## 5.2 Delete a Club



```
oop cw1 | src | Main | addRandomMatch
Project | Main | SchoolFootballClub.java | Match.java | CusComparator.java | LeagueManager.java | PremierLeagueManager.java | Main.java | PremierLeagueManagerTest.java
Run: Main
Press 1: Create a new football club and add it to premier league
Press 2: Delete an existing club from premier league
Press 3: Display statistics for club
Press 4: Display Premier League Table
Press 5: Add Played A Match
Press 6: Run GUI
1
Enter your club name:
PP
Enter your club location
PP
This club is successfully added to the league
-----
Object has been serialized
Press 1: Create a new football club and add it to premier league
Press 2: Delete an existing club from premier league
Press 3: Display statistics for club
Press 4: Display Premier League Table
Press 5: Add Played A Match
Press 6: Run GUI
2
Enter your club name:
PP
Deleted Successfully
Object has been serialized
Press 1: Create a new football club and add it to premier league
Press 2: Delete an existing club from premier league
Press 3: Display statistics for club
Press 4: Display Premier League Table
Press 5: Add Played A Match
Press 6: Run GUI
```

## 5.3 Add Played a match

```

oop cw1 | src | Main | addRandomMatch
Project | Project | SchoolFootballClub.java | Match.java | CusComparator.java | LeagueManager.java | PremierLeagueManager.java | Main.java | PremierLeagueManagerTest.java
Run: Main
uu
Enter your club location
uu
This club is successfully added to the league
-----
Object has been serialized
Press 1: Create a new football club and add it to premier league
Press 2: Delete an existing club from premier league
Press 3: Display statistics for club
Press 4: Display Premier League Table
Press 5: Add Played A Match
Press 6: Run GUI
5
Please Enter Date (format yyyy/mm/dd):
2021/01/03
Please Enter Home Team Name:
ss
Please Enter Away Team Name:
uu
Enter Home team goals:
8
Enter Away team goals:
5
-----
Object has been serialized
Press 1: Create a new football club and add it to premier league
Press 2: Delete an existing club from premier league
Press 3: Display statistics for club
Press 4: Display Premier League Table
Press 5: Add Played A Match
Press 6: Run GUI
|

```

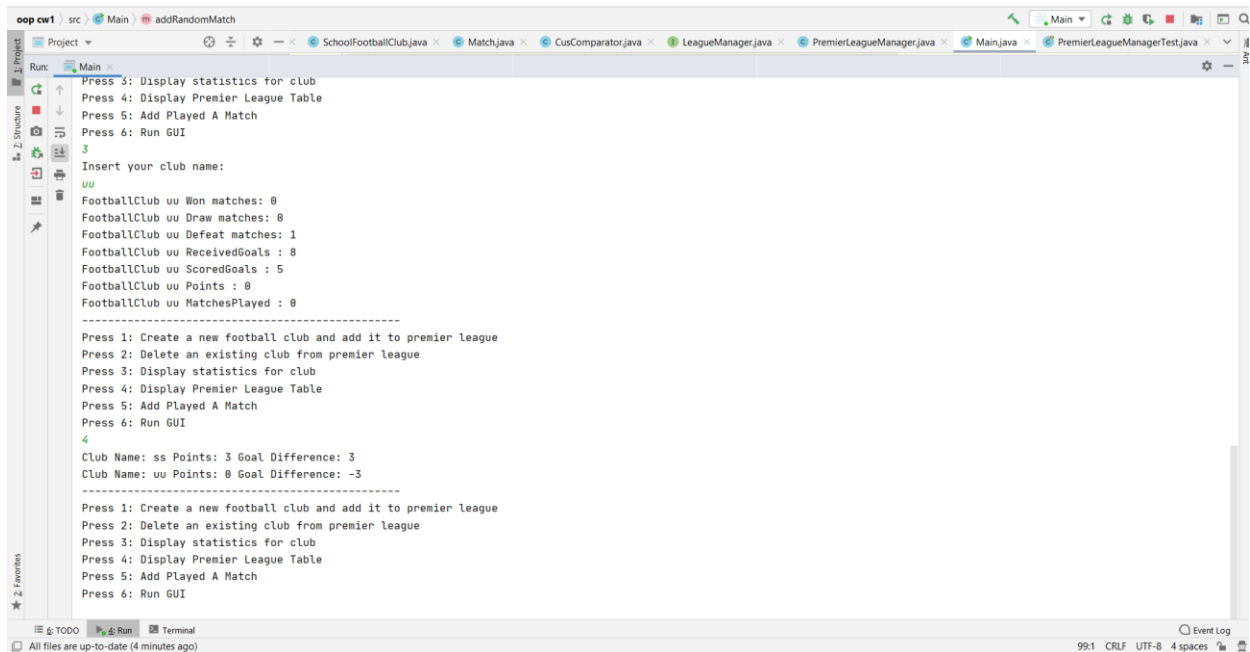
## 5.4 Display statistics for club

```

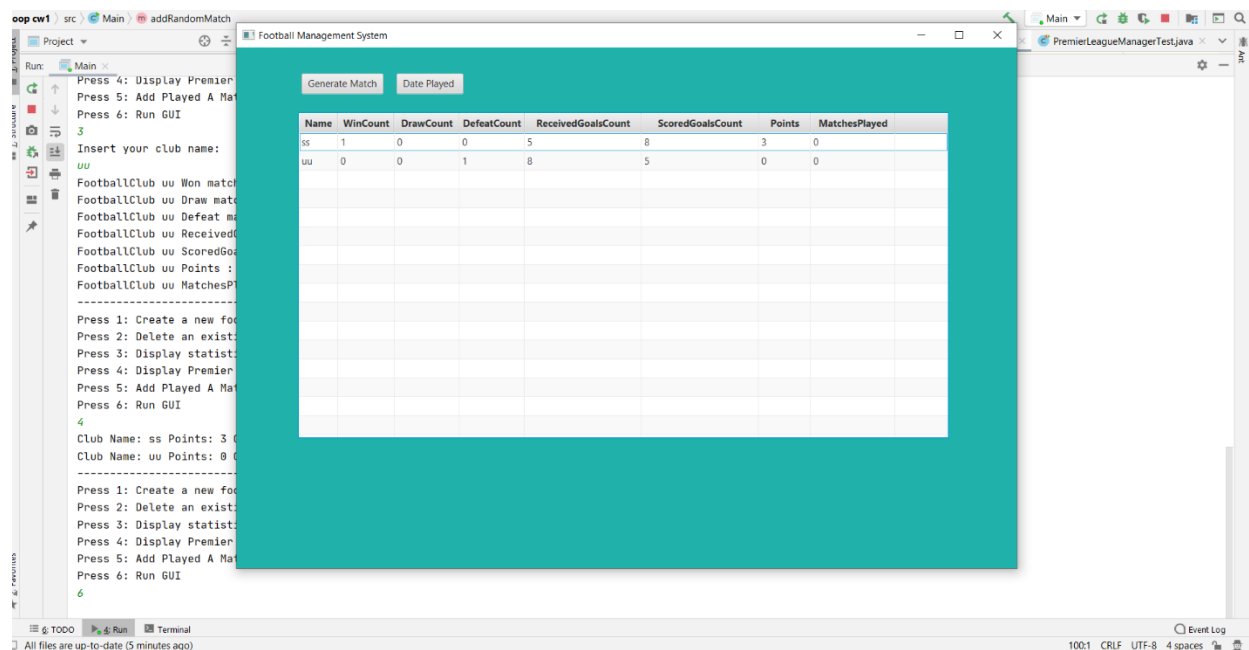
oop cw1 | src | Main | addRandomMatch
Project | Project | SchoolFootballClub.java | Match.java | CusComparator.java | LeagueManager.java | PremierLeagueManager.java | Main.java | PremierLeagueManagerTest.java
Run: Main
Press 4: Display Premier League Table
Press 5: Add Played A Match
Press 6: Run GUI
3
Insert your club name:
ss
FootballClub ss Won matches: 1
FootballClub ss Draw matches: 0
FootballClub ss Defeat matches: 0
FootballClub ss ReceivedGoals : 5
FootballClub ss ScoredGoals : 8
FootballClub ss Points : 3
FootballClub ss MatchesPlayed : 0
-----
Press 1: Create a new football club and add it to premier league
Press 2: Delete an existing club from premier league
Press 3: Display statistics for club
Press 4: Display Premier League Table
Press 5: Add Played A Match
Press 6: Run GUI
3
Insert your club name:
uu
FootballClub uu Won matches: 0
FootballClub uu Draw matches: 0
FootballClub uu Defeat matches: 1
FootballClub uu ReceivedGoals : 8
FootballClub uu ScoredGoals : 5
FootballClub uu Points : 0
FootballClub uu MatchesPlayed : 0
-----
Press 1: Create a new football club and add it to premier league

```

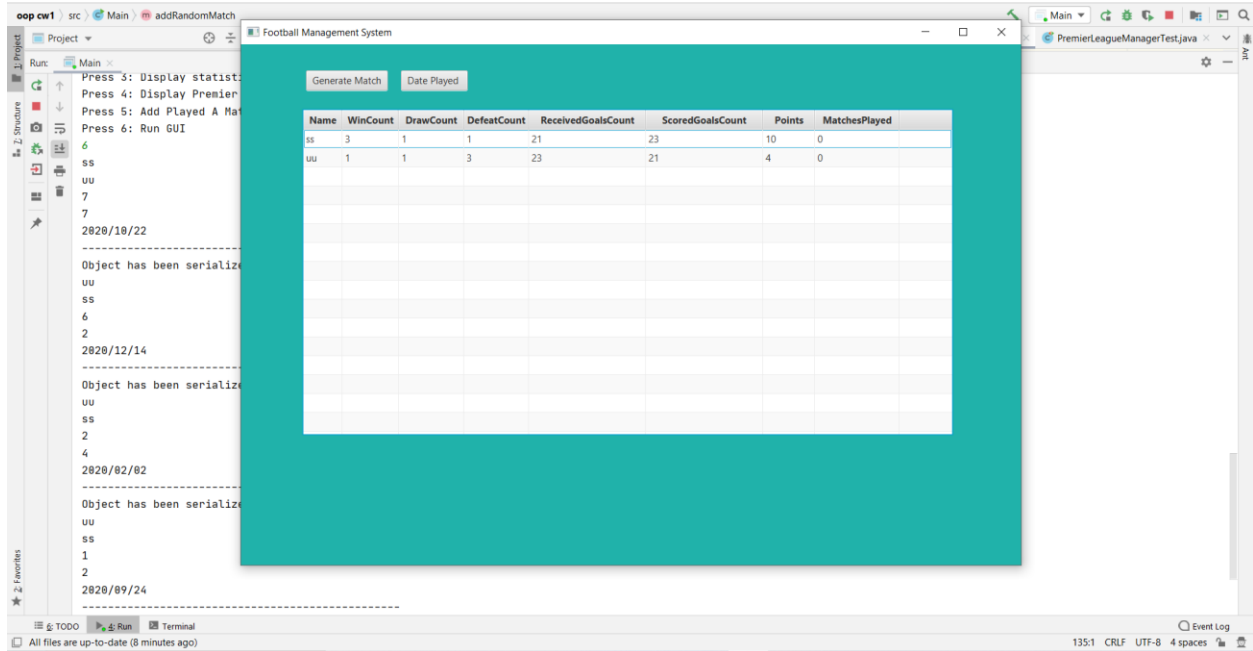
## 5.5 Display Premier League Table



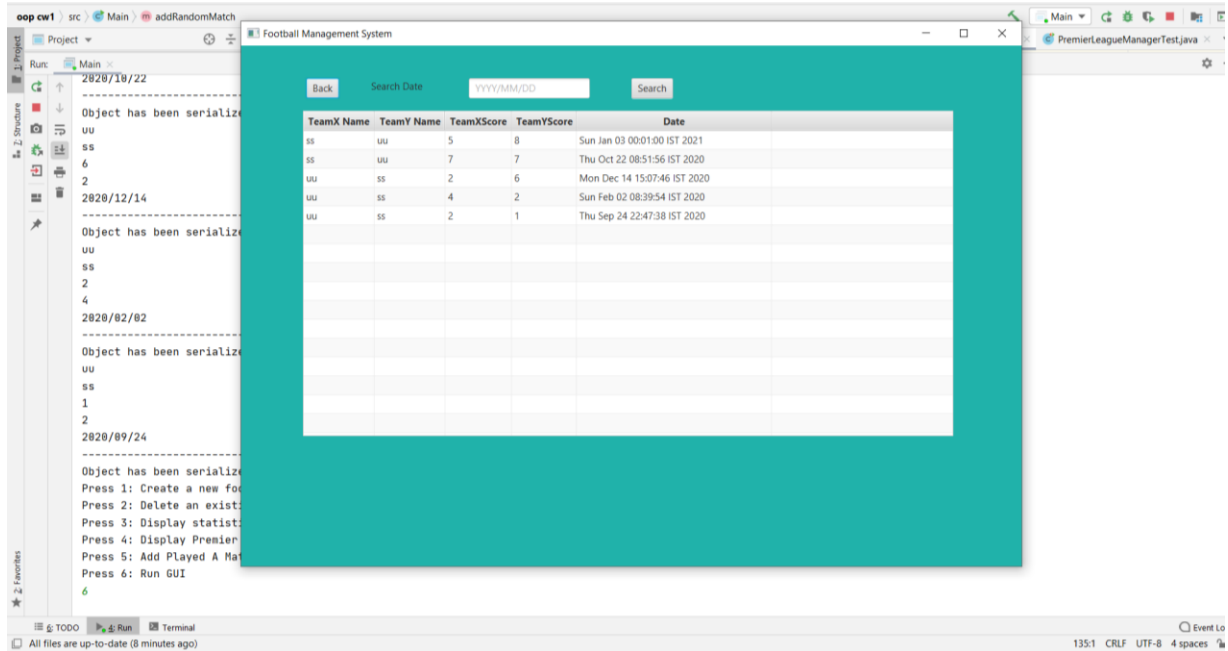
## 5.6 Display GUI



## 5.7 Randomly Generate match

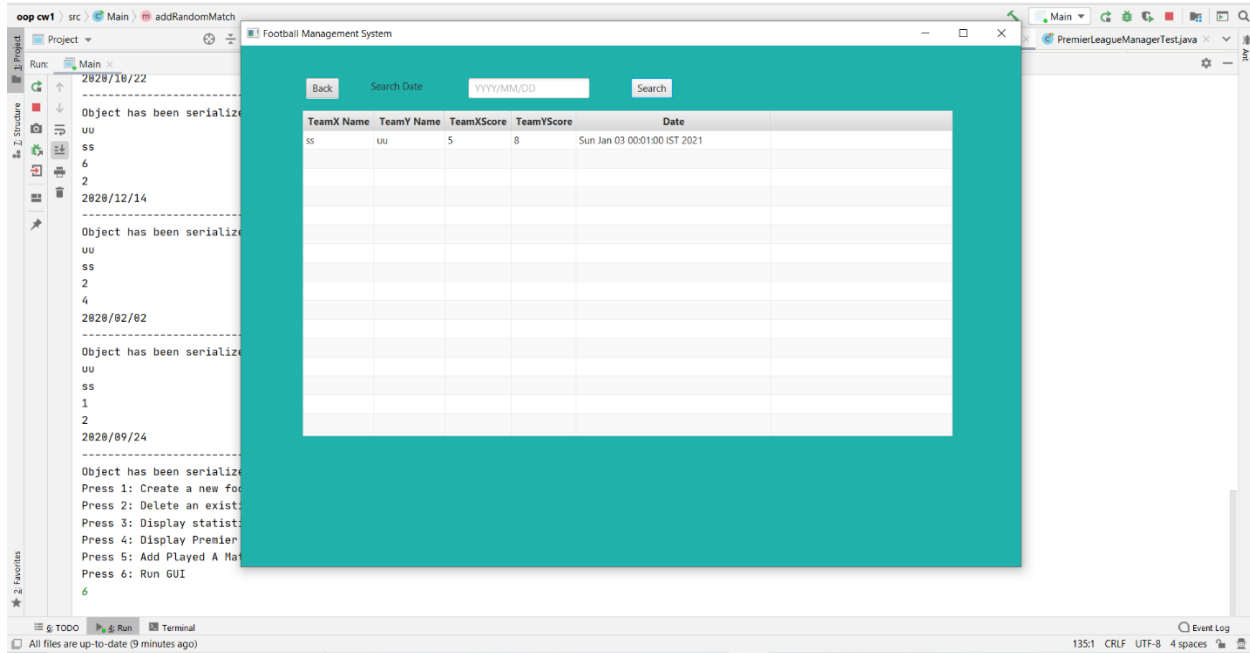


## 5.8 Date Played





## 5.9 Search date



## Modification for the initial plan and reasons

Added an option to search a match with the date played to make the system easy.

## Challenges, Solutions and Teamwork

During compilation and execution of the program, many errors were found. We worked together and solved those errors by sharing our knowledge and with the help of other resources such as internet. We divided the program within the members and created the whole program.

---

*Thank you!*

---