



INFORMATICS
INSTITUTE OF
TECHNOLOGY

UNIVERSITY OF
WESTMINSTER™

Informatics Institute of Technology
Department of Computing
(B.Eng.) in Software Engineering

Module: 5COSC007C.1 Object Oriented Programming

Module Leader: Guhanathan Poravi
Coursework

Student ID: 20191101

Student UOW ID: w1790354

Name: Kogula Kangaiveniyam

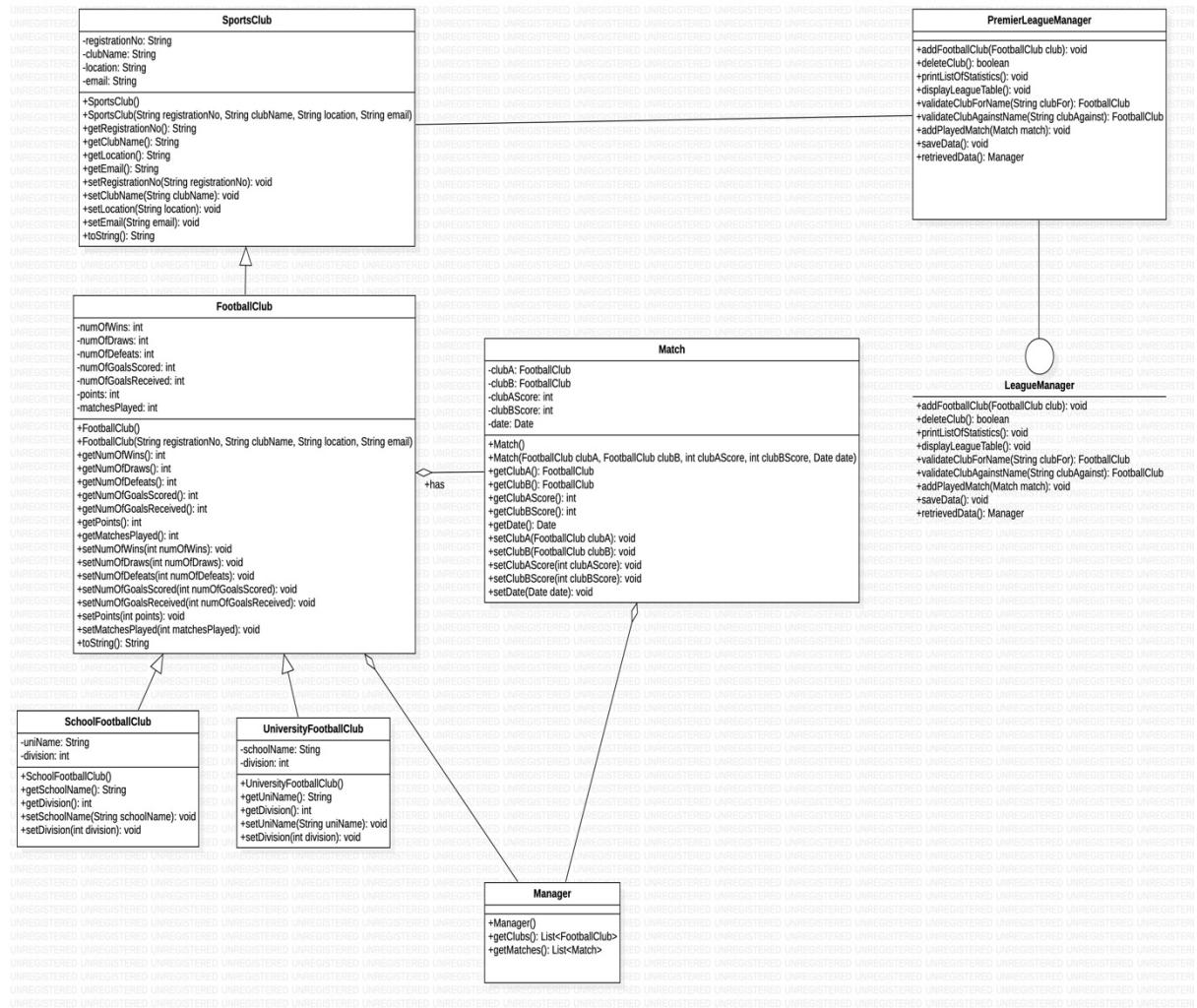
Group: M

Table of Contents

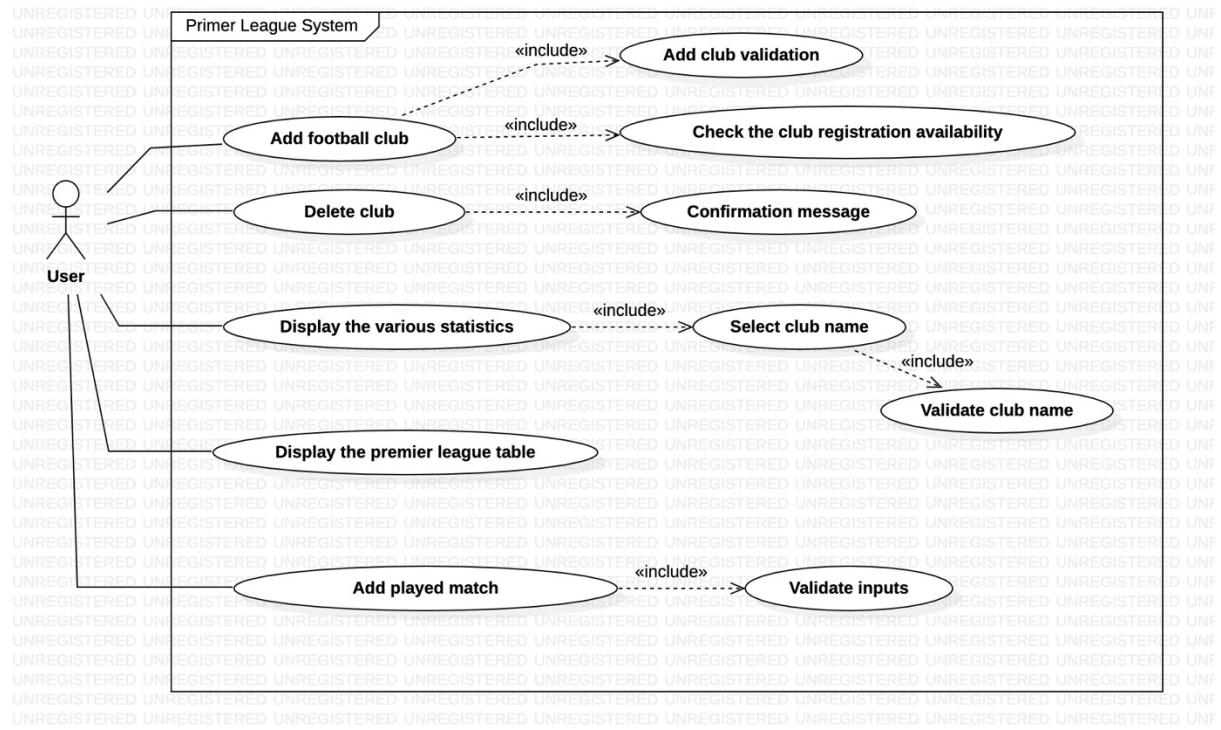
Design	1
Class Diagram.....	1
Use Case diagram for console.....	2
Use Case diagram for GUI.....	3
Q1)	3
SportsClub.java.....	3
FootballClub.java.....	4
UniversityFootballClub.java.....	5
SchoolFootballClub.java.....	6
Q2)	6
Match.java.....	6
ClubComparator.java	7
MatchForGUI.java	8
Manager.java.....	8
LeagueManager.java	9
PremierLeagueManager.java	9
W190354Application.java	14
Outputs	18
Validations	20
Q3)	21
LeagueService.java	21
LeagueServiceImpl.java.....	22
LeagueController.java.....	23
WebConfig.java	24
Angular part	24
Outputs	32

Design

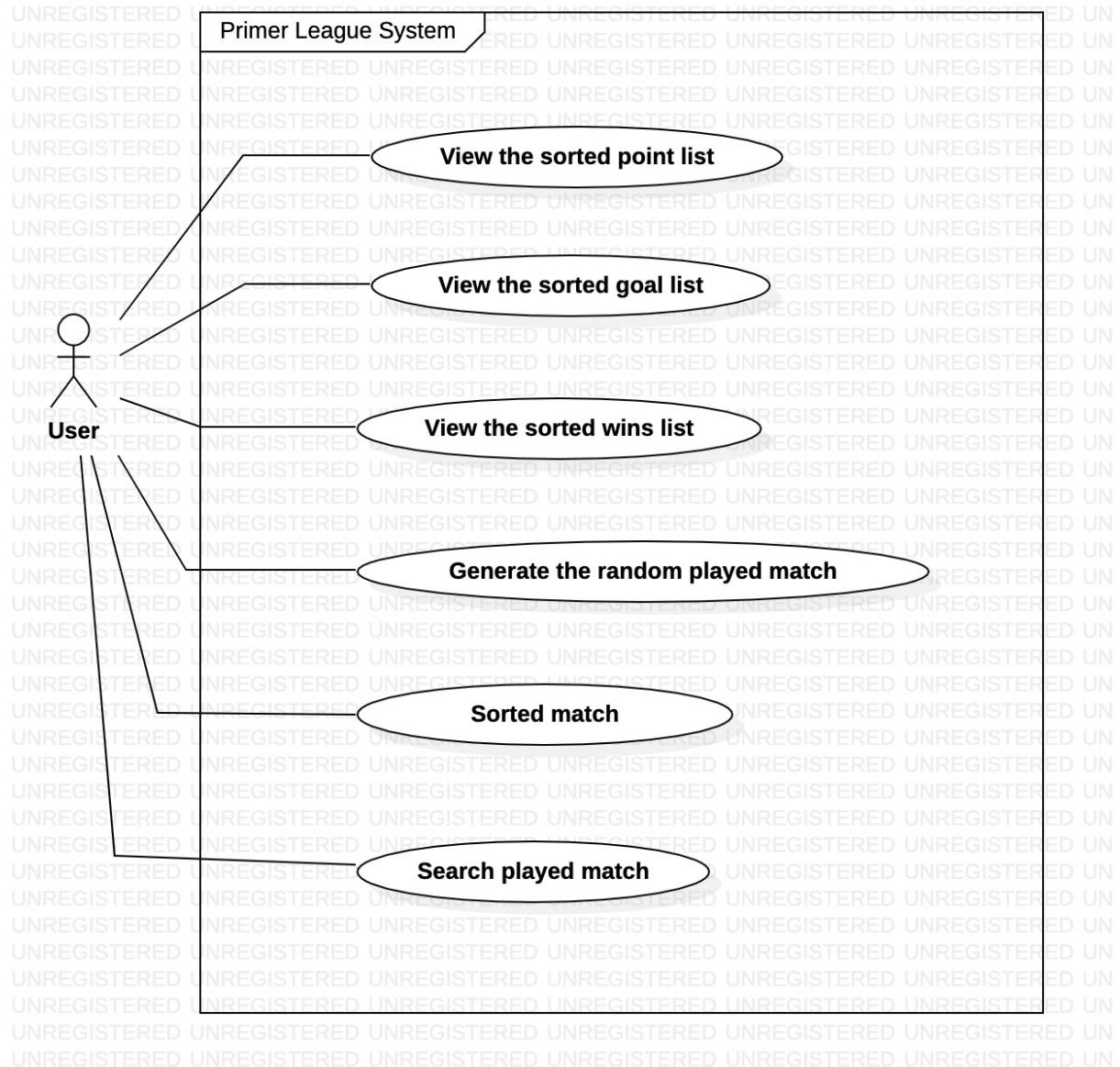
Class Diagram



Use Case diagram for console



Use Case diagram for GUI



Q1)

SportsClub.java

```
package lk.oop.cwk.w1790354.dto;  
/**  
 * Kogula Kangaiveniyam  
 * IIT No: 20191101  
 * UoW No: w1790354  
 */  
  
import lombok.Data;  
  
import java.io.Serializable;
```

```

@Data
public abstract class SportsClub implements Serializable {
    private static final long serialVersionUID = 777322596260401818L;

    private String registrationNo;
    private String clubName;
    private String location;
    private String email;

    public SportsClub(String registrationNo, String clubName, String
location, String email) {
        this.registrationNo = registrationNo;
        this.clubName = clubName;
        this.location = location;
        this.email = email;
    }

    public SportsClub() {
    }

    public SportsClub(String clubName) {
        this.clubName = clubName;
    }
}

```

FootballClub.java

```

package lk.oop.cwk.w1790354.dto;
/**
 * Kogula Kangaiveniyan
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lombok.Data;
import lombok.EqualsAndHashCode;

@EqualsAndHashCode(callSuper = true)
@Data
public class FootballClub extends SportsClub {
    private static final long serialVersionUID = 4458564001810445906L;

    private int numOfWins;
    private int numOfDraws;
    private int numOfDefeats;
    private int numOfGoalsScored;
    private int numOfGoalsReceived;
    private int points;
    private int matchesPlayed;

    public FootballClub(String registrationNo, String clubName, String
location, String email, int numOfWins, int numOfDraws, int numOfDefeats,
int numOfGoalsScored, int numOfGoalsReceived, int points, int
matchesPlayed) {
        super(registrationNo, clubName, location, email);
        this.numOfWins = numOfWins;
        this.numOfDraws = numOfDraws;
    }
}

```

```

        this.numOfDefeats = numOfDefeats;
        this.numOfGoalsScored = numOfGoalsScored;
        this.numOfGoalsReceived = numOfGoalsReceived;
        this.points = points;
        this.matchesPlayed = matchesPlayed;
    }

    public FootballClub(String registrationNo, String clubName, String location, String email) {
        super(registrationNo, clubName, location, email);
    }

    public FootballClub() {
    }
    public FootballClub(String clubName, int numOfWins, int numOfDefeats, int numOfDraws, int numOfGoalsScored, int numOfGoalsReceived, int points, int matchesPlayed ) {
        super(clubName);
        this.numOfWins = numOfWins;
        this.numOfDefeats = numOfDefeats;
        this.numOfDraws = numOfDraws;
        this.numOfGoalsScored= numOfGoalsScored;
        this.numOfGoalsReceived = numOfGoalsReceived;
        this.points = points;
        this.matchesPlayed = matchesPlayed;
    }
}

```

UniversityFootballClub.java

```

package lk.oop.cwk.w1790354.dto;
/**
 * Kogula Kangaiiveniyan
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lombok.Data;
import lombok.EqualsAndHashCode;

@EqualsAndHashCode(callSuper = true)
@Data
public class UniversityFootballClub extends FootballClub {
    private String uniName;
    private int division;

    public UniversityFootballClub(String registrationNo, String clubName, String location, String email, int numOfWins, int numOfDraws, int numOfDefeats, int numOfGoalsScored, int numOfGoalsReceived, int points, int matchesPlayed, String uniName, int division) {
        super(registrationNo, clubName, location, email, numOfWins, numOfDraws, numOfDefeats, numOfGoalsScored, numOfGoalsReceived, points, matchesPlayed);
        this.uniName = uniName;
        this.division = division;
    }

    public UniversityFootballClub(String uniName, int division) {

```

```

        this.uniName = uniName;
        this.division = division;

    }

    public UniversityFootballClub() {
    }

    public UniversityFootballClub(String registrationNo, String clubName,
String location, String email, String uniName, int division) {
        super(registrationNo, clubName, location, email);
        this.uniName = uniName;
        this.division = division;
    }

}

```

SchoolFootballClub.java

```

package lk.oop.cwk.w1790354.dto;
/**
 * Kogula Kangaiiveniyam
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lombok.Data;
import lombok.EqualsAndHashCode;

@EqualsAndHashCode(callSuper = true)
@Data
public class SchoolFootballClub extends FootballClub{
    private String schoolName;
    private int division;

    public SchoolFootballClub(String registrationNo, String clubName,
String location, String email, String schoolName, int division) {
        super(registrationNo, clubName, location, email);
        this.schoolName = schoolName;
        this.division = division;
    }
}

```

Q2)

Match.java

```

package lk.oop.cwk.w1790354.dto;
/**
 * Kogula Kangaiiveniyam
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lombok.Data;

```

```

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

@Data
public class Match implements Serializable {
    private FootballClub clubA;
    private FootballClub clubB;
    private int clubAScore;
    private int clubBScore;
    private Date date;

    public Match(FootballClub clubA, FootballClub clubB, int clubAScore,
    int clubBScore, Date date) {
        this.clubA = clubA;
        this.clubB = clubB;
        this.clubAScore = clubAScore;
        this.clubBScore = clubBScore;
        this.date = date;
    }

    public Match() {
    }
}

```

ClubComparator.java

```

package lk.oop.cwk.w1790354.dto;
/**
 * Kogula Kangaiveniyan
 * IIT No: 20191101
 * UoW No: w1790354
 */

import java.util.Comparator;

public class ClubComparator implements Comparator<FootballClub> {

    @Override
    public int compare(FootballClub club1, FootballClub club2) {

        if(club1.getPoints() > club2.getPoints())
            return -1;
        else if (club1.getPoints() < club2.getPoints())
            return 1;
        else {
            int goalDif = club1.getNumberOfGoalsScored() -
club1.getNumberOfGoalsReceived();
            int goalDif1 = club2.getNumberOfGoalsScored() -
club2.getNumberOfGoalsReceived();
            return Integer.compare(goalDif1, goalDif);
        }
    }
}

```

MatchForGUI.java

```
package lk.oop.cwk.w1790354.dto;
/**
 * Kogula Kangaiveniyam
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lombok.Data;
import java.text.SimpleDateFormat;

@Data
public class MatchForGUI {
    private String clubA;
    private String clubB;
    private int clubAScore;
    private int clubBScore;
    private String date;

    public MatchForGUI(Match obj) {
        this.clubA = obj.getClubA().getClubName();
        this.clubB = obj.getClubB().getClubName();
        this.clubAScore = obj.getClubAScore();
        this.clubBScore = obj.getClubBScore();
        this.date = new SimpleDateFormat("MM-dd-
yyyy").format(obj.getDate());
    }
}
```

Manager.java

```
package lk.oop.cwk.w1790354.dto;
/**
 * Kogula Kangaiveniyam
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lombok.Data;
import org.springframework.stereotype.Component;
import java.io.Serializable;
import java.util.List;

@Data
@Component
public class Manager implements Serializable {
    private static final long serialVersionUID = 4458564001810445904L;

    private List<FootballClub> clubs;
    private List<Match> matches;

    public Manager(List<FootballClub> clubs, List<Match> matches) {
        this.clubs = clubs;
    }
}
```

```

        this.matches = matches;
    }

    public Manager() {
    }
}

```

LeagueManager.java

```

package lk.oop.cwk.w1790354.service;
/**
 * Kogula Kangaiveniyan
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lk.oop.cwk.w1790354.dto.FootballClub;
import lk.oop.cwk.w1790354.dto.Manager;
import lk.oop.cwk.w1790354.dto.Match;

import java.io.IOException;

public interface LeagueManager {
    void addFootballClub(FootballClub club);
    boolean deleteClub(String clubName);
    void printListOfStatistics(String printClubName);
    void displayLeagueTable();
    FootballClub validateClubForName(String clubFor);
    FootballClub validateClubAgainstName(String clubAgainst);
    void addPlayedMatch(Match match);
    void saveData(String fileName) throws IOException;
    Manager retrievedData() throws IOException, ClassNotFoundException;

    void addRandomMatch() throws IOException;
}

```

PremierLeagueManager.java

```

package lk.oop.cwk.w1790354.service.impl;
/**
 * Kogula Kangaiveniyan
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lk.oop.cwk.w1790354.dto.*;
import lk.oop.cwk.w1790354.service.LeagueManager;
import org.springframework.stereotype.Service;

import java.io.*;
import java.util.*;
import java.util.concurrent.atomic.AtomicBoolean;
import java.util.stream.Collectors;

```

```

@Service
public class PremierLeagueManager implements LeagueManager {
    private List<FootballClub> clubList = new ArrayList<FootballClub>();
    //which is the I am coming to add the football club, delete club, sort,
    print, etc.
    private List<Match> matches = new ArrayList<Match>();
    private Manager manager = new Manager();

    @Override
    public void addFootballClub(FootballClub club) {
        if (clubList.size() <= 20) {
            Optional<FootballClub> clubExistOpt =
clubList.stream().filter(footballClub -> {
                return
footballClub.getClubName().equals(club.getClubName());
            }).findFirst();
            if (!clubExistOpt.isPresent()) {
                clubList.add(club);
                System.out.println(club.getClubName() + " is successfully
added to the primer league.");
            } else {
                System.out.println("There is a club with same name.");
            }
        } else {
            System.out.println("No free slots are available for new club
registration.");
        }
    }

    @Override
    public boolean deleteClub(String clubName) {
        boolean flag = false;
        for (FootballClub club : clubList) {
            if (club.getClubName().equals(clubName)) {
                flag = true;
                clubList.remove(club);
                System.out.println("Football Club with the club name: " +
clubName + " successfully removed.");
                System.out.println("No of occupied slots : " +
clubList.size());
                System.out.println("No of free slots : " + (20 -
clubList.size()));
                if (club instanceof UniversityFootballClub) { //what type
of has been deleted
                    System.out.println("Football Club type is:
UniversityFootballClub");
                } else {
                    System.out.println("Football Club type is:
SchoolFootballClub");
                }
                break;
            }
        }
        if (!flag) {
            System.out.println("Football Club with the club name: " +
clubName + " is not found.");
        }
        return flag;
    }

    @Override

```

```

public void printListOfStatistics(String printClubName) {
    for (FootballClub club : clubList) {
        if (club.getClubName().equals(printClubName)) {
            System.out.println("Club " + club.getClubName() + " matches
won: " + club.getNumOfWins());
            System.out.println("Club " + club.getClubName() + " matches
lost: " + club.getNumOfDefeats());
            System.out.println("Club " + club.getClubName() + " matches
draw: " + club.getNumOfDraws());
            System.out.println("Club " + club.getClubName() + " scored
goals: " + club.getNumOfGoalsScored());
            System.out.println("Club " + club.getClubName() + " "
received goals: " + club.getNumOfGoalsReceived());
            System.out.println("Club " + club.getClubName() + " points:
" + club.getPoints());
            System.out.println("Club " + club.getClubName() + " matches
played: " + club.getMatchesPlayed());
        }
        return;
    }
    System.out.println("No such club in premier league");
}

@Override
public void displayLeagueTable() {
    clubList.sort(new ClubComparator());
    for (FootballClub club : clubList) {
        System.out.println("Club: " + club.getClubName() + " Points: "
+ club.getPoints() + " goal difference: " + (club.getNumOfGoalsScored() -
club.getNumOfGoalsReceived()));
    }
}

@Override
public FootballClub validateClubForName(String clubFor) {
    FootballClub club1 = null;
    try {
        for (FootballClub club : clubList) {
            if ((club.getClubName().equals(clubFor))) {
                club1 = club;
                return club1;
            }
        }
    } catch (Exception e) {
        System.out.println(e);
    }
    return club1;
}

@Override
public FootballClub validateClubAgainstName(String clubAgainst) {
    FootballClub club2 = null;
    try {
        for (FootballClub club : clubList) {
            if ((club.getClubName().equals(clubAgainst))) {
                club2 = club;
                return club2;
            }
        }
    } catch (Exception e) {
        System.out.println(e);
    }
}

```

```

        }
        return club2;
    }

@Override
public void addPlayedMatch(Match match) {
    matches.add(match);
}

public void saveData(String fileName) throws IOException {
    fileName = "manager.txt";
    try {
        if (clubList.isEmpty()) {
            System.out.println("Premier League does not have any data
to save.");
            return;
        }

        FileOutputStream fos = new FileOutputStream(fileName);
        ObjectOutputStream oos = new ObjectOutputStream(fos);

        manager.setClubs(clubList);
        manager.setMatches(matches);
        oos.writeObject(manager);

        oos.flush();
        fos.close();
        oos.close();
        System.out.println("Data has been successfully saved to the
file.");
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}

@Override
public Manager retrievedData() throws IOException,
ClassNotFoundException {
    String fileName = "manager.txt";
    try {
        FileInputStream fis = new FileInputStream(fileName);
        ObjectInputStream ois = new ObjectInputStream(fis);

        for (; ; ) {
            try {
                manager = (Manager) ois.readObject();
                clubList = manager.getClubs();
                matches = manager.getMatches();
            } catch (EOFException e) {
                break;
            }
        }

        fis.close();
        ois.close();
        //System.out.println("Club date : "+ matches.get(0).getDate());
        System.out.println("Data has been successfully loaded from the
file.");
    } catch (FileNotFoundException e) {
}

```

```

        }
        return manager;
    }

@Override
public void addRandomMatch() throws IOException {
    Random random = new Random();
    FootballClub clubFor =
clubList.get(random.nextInt(clubList.size()));
    clubList.remove(clubFor);
    FootballClub clubAgainst =
clubList.get(random.nextInt(clubList.size()));
    int clubForGoals = random.nextInt(10);
    int clubAgainstGoals = random.nextInt(10);
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(new Date());
    calendar.add(Calendar.DATE, random.nextInt(10));
    Date date= calendar.getTime();

    Match match = new Match();
    match.setDate(date);
    match.setClubA(clubFor);
    match.setClubB(clubAgainst);
    match.setClubAScore(clubForGoals);
    match.setClubBScore(clubAgainstGoals);
    clubFor.setNumOfGoalsScored(clubFor.getNumOfGoalsScored() +
clubForGoals);
    clubAgainst.setNumOfGoalsScored(clubAgainst.getNumOfGoalsScored() +
clubAgainstGoals);
    clubFor.setNumOfGoalsReceived(clubFor.getNumOfGoalsReceived() +
clubAgainstGoals);

    clubAgainst.setNumOfGoalsReceived(clubAgainst.getNumOfGoalsReceived() +
clubForGoals);
    clubFor.setMatchesPlayed(clubFor.getMatchesPlayed() + 1);
    clubAgainst.setMatchesPlayed(clubAgainst.getMatchesPlayed() + 1);

    if (clubForGoals > clubAgainstGoals) {
        clubFor.setPoints(clubFor.getPoints() + 3);
        clubFor.setNumOfWins(clubFor.getNumOfWins() + 1);
        clubAgainst.setNumOfDefeats(clubAgainst.getNumOfDefeats() + 1);
    } else if (clubForGoals < clubAgainstGoals) {
        clubAgainst.setPoints(clubAgainst.getPoints() + 3);
        clubAgainst.setNumOfWins(clubAgainst.getNumOfWins() + 1);
        clubFor.setNumOfDefeats(clubFor.getNumOfDefeats() + 1);
    } else {
        clubFor.setPoints(clubFor.getPoints() + 1);
        clubAgainst.setPoints(clubAgainst.getPoints() + 1);
        clubFor.setNumOfDraws(clubFor.getNumOfDraws() + 1);
        clubAgainst.setNumOfDraws(clubAgainst.getNumOfDraws() + 1);
    }
    this.addPlayedMatch(match);
    this.replaceClub(clubFor);
    this.replaceClub(clubAgainst);
    this.saveData("manager.txt");
}

private void replaceClub(FootballClub club) {
    AtomicBoolean clubExists = new AtomicBoolean(false);
    clubList = clubList.stream().map(footballClub -> {
        if(footballClub.getClubName().equals(club.getClubName())) {

```

```

        clubExists.set(true);
        return club;
    }
    return footballClub;
}).collect(Collectors.toList());
if(!clubExists.get()){
    clubList.add(club);
}
}
}
}

```

W190354Application.java

```

package lk.oop.cwk.w1790354;
/**
 * Kogula Kangaiveniyam
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lk.oop.cwk.w1790354.dto.FootballClub;
import lk.oop.cwk.w1790354.dto.Match;
import lk.oop.cwk.w1790354.dto.SchoolFootballClub;
import lk.oop.cwk.w1790354.dto.UniversityFootballClub;
import lk.oop.cwk.w1790354.service.impl.PremierLeagueManager;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

@SpringBootApplication
public class W1790354Application {

    private final static PremierLeagueManager manager = new
PremierLeagueManager();
    private static int count = 0;

    public static void main(String[] args) throws IOException,
ClassNotFoundException {
        SpringApplication.run(W1790354Application.class, args);
        manager.retrievedData();

        while (true) {
            System.out.println("*****      Premier League menu      *****\n"
+
"                                " + "press 1: Create new club and add it to
league.....\n" +
"                                " + "press 2: Delete existing club from
league.....\n" +
"                                " + "press 3: Display Statistics for
club.....\n" +
"                                " + "press 4: Display the Premier League
Table.....\n" +
"                                " + "press 5: Add a Played
Match.....\n" +
"                                " + "press 6: Save
"
        }
    }
}

```

```

Data.....\n" +
"           press 7:\n"
Exit.....\n" +
"Enter Input : ");

Scanner input = new Scanner(System.in);
int choice = input.nextInt();

switch (choice) {
    case 1:
        addClub();
        break;
    case 2:
        deleteClub();
        break;
    case 3:
        printListOfStatistics();
        break;
    case 4:
        manager.displayLeagueTable();
        break;
    case 5:
        addPlayedMatch();
        break;
    case 6:
        manager.saveData("manager.txt");
        System.out.println("* Saved *");
        break;
    case 7:
        System.out.println("* Exit *");
        System.exit(0);
    default:
        System.out.println("Wrong Choice!! Please re-
enter..."); }
}

private static void addClub() {
    Scanner input = new Scanner(System.in);
    if (count <= 20) {
        System.out.println("Registration No: ");
        String registrationNo = input.nextLine();

        System.out.println("Insert Club Name: ");
        String clubName = input.nextLine();

        System.out.println("Insert Club Location: ");
        String location = input.nextLine();

        System.out.println("Email Address: ");
        String email = input.nextLine();

        System.out.println("Enter the type of Football Club (U-
University Football Club/ S- School Football Club) : ");
        String type = input.nextLine();
        FootballClub club = new FootballClub();

        switch (type) {
            case "U":
            case "u":
```

```

        System.out.println("Enter the University name: ");
        String uniName = input.nextLine();
        System.out.println("Enter the division: ");
        int division = input.nextInt();
        club = new UniversityFootballClub(registrationNo,
clubName, location, email, uniName, division);
        break;

    case "S":
    case "s":
        System.out.println("Enter the School name: ");
        String schoolName = input.nextLine();
        System.out.println("Enter the division: ");
        division = input.nextInt();
        club = new SchoolFootballClub(registrationNo, clubName,
location, email, schoolName, division);
        break;

    default:
        System.out.println("Invalid Input");
    }
    manager.addFootballClub(club);
    count++;
} else {
    System.out.println("no free slots"); //Club registration is
above 20 limits
}
}

private static void deleteClub() {
Scanner input = new Scanner(System.in);
System.out.println("Enter the club name : ");
String clubName = input.nextLine();
System.out.println("Do you want Delete this Record y/n ");
String choice = input.next();
if (choice.equalsIgnoreCase("y")) {
    boolean result = manager.deleteClub(clubName);
    if (result) {
        count--;
    }
} else {
    System.out.println("Delete operation is failed.");
}
}

private static void printListOfStatistics() {
Scanner sc = new Scanner(System.in);
System.out.println("Enter the club name: ");
String printClubName = sc.nextLine();
manager.printListOfStatistics(printClubName);
}

private static void addPlayedMatch() {
Scanner sc = new Scanner(System.in);
System.out.println("Enter date (format MM-DD-YYYY): ");
String playedDate = sc.nextLine();
Date date;
try {
    date = new SimpleDateFormat("MM-dd-yyyy").parse(playedDate);
} catch (ParseException ex) {
    System.out.println("You have to enter date in format MM-DD-

```

```

        YYYY");
            return;
        }

        System.out.println("Enter Club For: ");
        String clubFor = sc.nextLine();
        FootballClub club1 = manager.validateClubForName(clubFor);
        //System.out.println(club1.getClubName());

        System.out.println("Enter Club Against: ");
        String clubAgainst = sc.nextLine();
        FootballClub club2 = manager.validateClubAgainstName(clubAgainst);

        if (club1 == null || club2 == null) {
            System.out.println("No such club in premier league");
            return;
        }

        System.out.println("Enter Club For goals: ");
        String line = sc.nextLine();
        int clubForGoals = -1;
        try {
            clubForGoals = Integer.parseInt(line);
        } catch (Exception e) {
        }
        if (clubForGoals == -1) {
            System.out.println("You have to enter number of goals");
            return;
        }

        System.out.println("Enter Club Against goals: ");
        line = sc.nextLine();
        int clubAgainstGoals = -1;
        try {
            clubAgainstGoals = Integer.parseInt(line);
        } catch (Exception e) {
        }
        if (clubAgainstGoals == -1) {
            System.out.println("You have to enter number of goals");
            return;
        }

        Match match = new Match();
        match.setDate(date);
        match.setClubA(club1);
        match.setClubB(club2);
        match.setClubAScore(clubAgainstGoals);
        match.setClubBScore(clubForGoals);
        club1.setNumOfGoalsScored(club1.getNumOfGoalsScored() +
clubForGoals);
        club2.setNumOfGoalsScored(club2.getNumOfGoalsScored() +
clubAgainstGoals);
        club1.setNumOfGoalsReceived(club1.getNumOfGoalsReceived() +
clubAgainstGoals);
        club2.setNumOfGoalsReceived(club2.getNumOfGoalsReceived() +
clubForGoals);
        club1.setMatchesPlayed(club1.getMatchesPlayed() + 1);
        club2.setMatchesPlayed(club2.getMatchesPlayed() + 1);

        if (clubForGoals > clubAgainstGoals) {
            club1.setPoints(club1.getPoints() + 3);
        }
    }
}

```

```
        club1.setNumOfWins(club1.getNumOfWins() + 1);
        club2.setNumOfDefeats(club2.getNumOfDefeats() + 1);
    } else if (clubForGoals < clubAgainstGoals) {
        club2.setPoints(club2.getPoints() + 3);
        club2.setNumOfWins(club2.getNumOfWins() + 1);
        club1.setNumOfDefeats(club1.getNumOfDefeats() + 1);
    } else {
        club1.setPoints(club1.getPoints() + 1);
        club2.setPoints(club2.getPoints() + 1);
        club1.setNumOfDraws(club1.getNumOfDraws() + 1);
        club2.setNumOfDraws(club2.getNumOfDraws() + 1);
    }
    manager.addPlayedMatch(match);
}
}
```

Outputs

```
w1790354 — java -jar ~/OOP_Tutorials/w1790354/target/w1790354-0.0.1-SNAPSHOT.jar — 120x29
[INFO] [tomcat] Tomcat initialized with port(s): 8080 (http)
[INFO] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
[INFO] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
[INFO] [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.41]
[INFO] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
[INFO] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1177 ms
[INFO] [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
[INFO] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
[INFO] [main] lk.oop.cwk.w1790354.W1790354Application : Started W1790354Application in 17.323 seconds (JVM running for 17.775)
[INFO] [main] lk.oop.cwk.w1790354.W1790354Application : Data has been successfully loaded from the file.

***** Premier League menu *****
press 1: Create new club and add it to league.....
press 2: Delete existing club from league.....
press 3: Display Statistics for club.....
press 4: Display the Premier League Table.....
press 5: Add a Played Match.....
press 6: Save Data.....
press 7: Exit.....
```

```
***** Premier League menu *****  

press 1: Create new club and add it to league.....  

press 2: Delete existing club from league.....  

press 3: Display Statistics for club.....  

press 4: Display the Premier League Table.....  

press 5: Add a Played Match.....  

press 6: Save Data.....  

press 7: Exit.....  

Enter Input :  

1  

Registration No:  

5092  

Insert Club Name:  

Everton  

Insert Club Location:  

UK  

Email Address:  

everking@gmail.com  

Enter the type of Football Club (U- University Football Club/ S- School Football Club) :  

u  

Enter the University name:  

Wesminister  

Enter the division:  

2  

Everton is successfully added to the primer league.
```

```
***** Premier League menu *****  

press 1: Create new club and add it to league.....  

press 2: Delete existing club from league.....  

press 3: Display Statistics for club.....  

press 4: Display the Premier League Table.....  

press 5: Add a Played Match.....  

press 6: Save Data.....  

press 7: Exit.....  

Enter Input :  

2  

Enter the club name :  

Everton  

Do you want Delete this Record y/n  

y  

Football Club with the club name: Everton successfully removed.  

No of occupied slots : 4  

No of free slots : 16  

Football Club type is: UniversityFootballClub
```

```
***** Premier League menu *****  

press 1: Create new club and add it to league.....  

press 2: Delete existing club from league.....  

press 3: Display Statistics for club.....  

press 4: Display the Premier League Table.....  

press 5: Add a Played Match.....  

press 6: Save Data.....  

press 7: Exit.....  

Enter Input :  

3  

Enter the club name:  

Arsenal  

Club Arsenal matches won: 2  

Club Arsenal matches lost: 0  

Club Arsenal matches draw: 0  

Club Arsenal scored goals: 12  

Club Arsenal received goals: 3  

Club Arsenal points: 6  

Club Arsenal matches played: 2
```

```

*****      Premier League menu      *****
press 1: Create new club and add it to league.....
press 2: Delete existing club from league.....
press 3: Display Statistics for club.....
press 4: Display the Premier League Table.....
press 5: Add a Played Match.....
press 6: Save Data.....
press 7: Exit.....
Enter Input :
4
Club: Arsenal Points: 6 goal difference: 9
Club: Man Cir=ty Points: 3 goal difference: 6
Club: Man Utd Points: 0 goal difference: -6
Club: Liverpool Points: 0 goal difference: -9

```

```

*****      Premier League menu      *****
press 1: Create new club and add it to league.....
press 2: Delete existing club from league.....
press 3: Display Statistics for club.....
press 4: Display the Premier League Table.....
press 5: Add a Played Match.....
press 6: Save Data.....
press 7: Exit.....
Enter Input :
5
Enter date (format MM-DD-YYYY):
01-04-2021
Enter Club For:
Arsenal
Enter Club Against:
Man Cir=ty
Enter Club For goals:
5
Enter Club Against goals:
4

```

```

*****      Premier League menu      *****
press 1: Create new club and add it to league.....
press 2: Delete existing club from league.....
press 3: Display Statistics for club.....
press 4: Display the Premier League Table.....
press 5: Add a Played Match.....
press 6: Save Data.....
press 7: Exit.....
Enter Input :
7
* Exit          *

```

Validations

```

Insert Club Name:
Arsenal
Insert Club Location:
uk
Email Address:
gvhv@gmail.com
Enter the type of Football Club (U- University Football Club/ S- School Football Club) :
s
Enter the School name:
rc
Enter the division:
1
There is a club with same name.

```

```

*****      Premier League menu      *****
press 1: Create new club and add it to league.....
press 2: Delete existing club from league.....
press 3: Display Statistics for club.....
press 4: Display the Premier League Table.....
press 5: Add a Played Match.....
press 6: Save Data.....
press 7: Exit.....
Enter Input :
2
Enter the club name :
Everton
Do you want Delete this Record y/n
y
Football Club with the club name: Everton is not found.

```

```

*****      Premier League menu      *****
press 1: Create new club and add it to league.....
press 2: Delete existing club from league.....
press 3: Display Statistics for club.....
press 4: Display the Premier League Table.....
press 5: Add a Played Match.....
press 6: Save Data.....
press 7: Exit.....
Enter Input :
3
Enter the club name:
Everton
No such club in premier league

```

```

*****      Premier League menu      *****
press 1: Create new club and add it to league.....
press 2: Delete existing club from league.....
press 3: Display Statistics for club.....
press 4: Display the Premier League Table.....
press 5: Add a Played Match.....
press 6: Save Data.....
press 7: Exit.....
Enter Input :
5
Enter date (format MM-DD-YYYY):
09/10/2020
You have to enter date in format MM-DD-YYYY

```

Q3)

LeagueService.java

```

package lk.oop.cwk.w1790354.service;
/**
 * Kogula Kangaiveniyan
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lk.oop.cwk.w1790354.dto.FootballClub;
import lk.oop.cwk.w1790354.dto.MatchForGUI;

```

```

import java.io.IOException;
import java.util.List;

public interface LeagueService {

    List<FootballClub> getClubs() throws IOException,
ClassNotFoundException;

    List<MatchForGUI> getMatches() throws IOException,
ClassNotFoundException;

    List<MatchForGUI> generateMatch() throws IOException,
ClassNotFoundException;
}

```

LeagueServiceImpl.java

```

package lk.oop.cwk.w1790354.service.impl;
/**
 * Kogula Kangaiiveniyam
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lk.oop.cwk.w1790354.dto.FootballClub;
import lk.oop.cwk.w1790354.dto.Manager;
import lk.oop.cwk.w1790354.dto.MatchForGUI;
import lk.oop.cwk.w1790354.service.LeagueManager;
import lk.oop.cwk.w1790354.service.LeagueService;
import org.springframework.stereotype.Service;

import java.io.IOException;
import java.util.List;
import java.util.stream.Collectors;

@Service
public class LeagueServiceImpl implements LeagueService {

    private final LeagueManager leagueManager;

    public LeagueServiceImpl(LeagueManager leagueManager) {
        this.leagueManager = leagueManager;
    }

    @Override
    public List<FootballClub> getClubs() throws IOException,
ClassNotFoundException {
        Manager manager = leagueManager.retrievedData();
        return manager.getClubs();
    }

    @Override
    public List<MatchForGUI> getMatches() throws IOException,
ClassNotFoundException {
        Manager manager = leagueManager.retrievedData();
        return getMatchesForGUI(manager);
    }
}

```

```

    }

    @Override
    public List<MatchForGUI> generateMatch() throws IOException,
    ClassNotFoundException {
        leagueManager.addRandomMatch();
        return getMatchesForGUI(leagueManager.retrievedData());
    }

    private List<MatchForGUI> getMatchesForGUI(Manager manager) {
        return manager.getMatches().stream().map(match -> new
MatchForGUI(match)).collect(Collectors.toList());
    }
}

```

LeagueController.java

```

package lk.oop.cwk.w1790354.controller;
/**
 * Kogula Kangaiveniyan
 * IIT No: 20191101
 * UoW No: w1790354
 */

import lk.oop.cwk.w1790354.dto.FootballClub;
import lk.oop.cwk.w1790354.dto.MatchForGUI;
import lk.oop.cwk.w1790354.service.LeagueService;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import java.io.IOException;
import java.util.List;

@RestController
@RequestMapping("/")
public class LeagueController {

    private final LeagueService leagueService;

    public LeagueController(LeagueService leagueService) {
        this.leagueService = leagueService;
    }

    @GetMapping("/clubs")
    public List<FootballClub> getClubs() throws IOException,
    ClassNotFoundException {
        return leagueService.getClubs();
    }

    @GetMapping("/matches")
    public List<MatchForGUI> getMatches() throws IOException,
    ClassNotFoundException {
        return leagueService.getMatches();
    }

    @GetMapping("/generate")
    public List<MatchForGUI> generateMatch() throws IOException,
    ClassNotFoundException {
        return leagueService.generateMatch();
    }
}

```

```
}
```

WebConfig.java

```
package lk.oop.cwk.w1790354.config;
/**
 * Kogula Kangaiiveniyan
 * IIT No: 20191101
 * UoW No: w1790354
 */

import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.CorsRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration
public class WebConfig implements WebMvcConfigurer {

    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping("/**")
            .allowedMethods("GET", "POST", "PUT", "DELETE")
            .allowedOrigins("*")
            .allowedHeaders("*");
    }
}
```

Angular part

index.html

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>PrimerLeague</title>
    <base href="/">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
    <app-root></app-root>
</body>
</html>
```

app.component.html

```
<h1 class="text-center"> {{title}} </h1>
<div class = "container" style="color: aliceblue;">
    <body class="text-center">
```

```
<app-header></app-header>
</body>
</div>
```

app.component.css

```
body {
  background-image: url("../app/wp6742461.jpg");
  background-repeat: no-repeat;
  background-position: center;
  width: 100%;
  height: 700px;
}
```

app.component.ts

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Primer League Table';
}
```

app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { ReactiveFormsModule } from '@angular/forms';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { FootballPointsListComponent } from './football-points-list/football-
points-list.component';
import { HeaderComponent } from './header/header.component';
import { OrderModule } from 'ngx-order-pipe';
import { SortMatchListComponent } from './sort-match-list/sort-match-
list.component';
import { FilterPipeModule } from 'ngx-filter-pipe';

@NgModule({
  declarations: [
    AppComponent,
    FootballPointsListComponent,
```

```

    HeaderComponent,
    SortMatchListComponent,
],
imports: [
  BrowserModule,
  AppRoutingModule,
  OrderModule,
  FilterPipeModule,
  FormsModule,
  ReactiveFormsModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }

```

app-routing.module.ts

```

/**
 * Kogula Kangaiveniyan
 * IIT No: 20191101
 * UoW No: w1790354
 */
import { Component, NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { FootballPointsListComponent } from './football-points-list/football-
points-list.component';
import { SortMatchListComponent } from './sort-match-list/sort-match-
list.component';

const routes: Routes = [
  {
    path: 'table',
    component: FootballPointsListComponent
  },
  {
    path: 'matches',
    component: SortMatchListComponent
  }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }

```

sports-club.ts

```
export class SportsClub {  
    registrationNo: number;  
    clubName: string;  
    location: string;  
    email: string;  
}
```

football-club.ts

```
import { SportsClub } from "./sports-club";  
  
export class FootballClub extends SportsClub {  
    numOfWins: number;  
    numOfDraws: number;  
    numOfDefeats: number;  
    numOfGoalsScored: number;  
    numOfGoalsReceived: number;  
    points: number;  
    matchesPlayed: number;  
}
```

match.ts

```
export class Match {  
    clubA: string;  
    clubB: string;  
    clubAScore: number;  
    clubBScore: number;  
    date: string;  
}
```

header.component.html

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">  
    <div class="collapse navbar-collapse" id="navbarNav">  
        <ul class="navbar-nav">  
            <li class="nav-item active">  
                <a class="nav-link" (click) = "goPoints()">Club Table</a>  
            </li>  
            <li class="nav-item">  
                <a class="nav-link" (click) = "goMatches()">Matches</a>  
            </li>  
        </ul>  
    </div>  
</nav>  
<router-outlet></router-outlet>
```

header.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';

@Component({
  selector: 'app-header',
  templateUrl: './header.component.html',
  styleUrls: ['./header.component.css']
})
export class HeaderComponent implements OnInit {

  constructor(private router: Router) { }

  goPoints(){
    this.router.navigate(['table']);
  }

  goMatches(){
    this.router.navigate(['matches']);
  }

  ngOnInit(): void {
  }
}
```

football-points-list.component.html

```
<table class = "table table-striped" style="color: aliceblue;">
  <thead>
    <tr>
      <th>Club Name</th>
      <th>Wins</th>
      <th>Lost</th>
      <th>Draw</th>
      <th>Scored Goals</th>
      <th>Received Goals</th>
      <th>Points</th>
      <th>Played Matches</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor = "let footballClub of football.sort().reverse()">
      <td> {{footballClub.clubName }}</td>
      <td> {{footballClub.numOfWins }} </td>
      <td> {{footballClub.numOfDefeats }} </td>
      <td> {{footballClub.numOfDraws }} </td>
      <td> {{footballClub.numOfGoalsScored }} </td>
      <td> {{footballClub.numOfGoalsReceived }} </td>
      <td> {{footballClub.points }} </td>
    </tr>
  </tbody>
</table>
```

```

        <td> {{footballClub.matchesPlayed}} </td>

    </tr>
</tbody>
</table>

<button class="btn btn-dark btn-lg active"
(click)="sortBy('points')">Points</button>
<button class="btn btn-dark btn-lg active"
(click)="sortBy('numOfGoalsScored')">Goals</button>
<button class="btn btn-dark btn-lg active"
(click)="sortBy('numOfWins')">Wins</button>
```

football-points-list.component.css

```
button{
    margin: 13px 12px 12px 10px;
}
```

football-points-list.component.ts

```

import { Component, OnInit } from '@angular/core';
import { OrderPipe } from 'ngx-order-pipe';
import axios from 'axios';

@Component({
  selector: 'app-football-points-list',
  templateUrl: './football-points-list.component.html',
  styleUrls: ['./football-points-list.component.css']
})
export class FootballPointsListComponent implements OnInit {

  football = [];

  clublist = [];
  order: string = 'points';

  constructor(private orderPipe: OrderPipe) {}

  ngOnInit(): void {
    axios.get("http://localhost:8080/clubs").then(resp => {
      this.clublist = resp.data;
      this.football = this.orderPipe.transform(this.clublist, this.order);
    });

    console.log(this.football);
  }
}
```

```

sortBy(orderString){
  console.log(orderString);
  this.football = this.orderPipe.transform(this.clublist, orderString);
}
}

```

sort-match-list.component.html

```

<h4 style="color: aliceblue;">Date Wise Sorting List</h4>
<table class = "table table-striped" style="color: aliceblue;">
  <thead>
    <tr>
      <th>Club For</th>
      <th>Score For</th>
      <th>Club Against</th>
      <th>Score Against</th>
      <th>Date</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor = "let match of matchArray">
      <td> {{match.clubA }}</td>
      <td> {{match.clubAScore }} </td>
      <td> {{match.clubB }} </td>
      <td> {{match.clubBScore }} </td>
      <td> {{match.date }} </td>
    </tr>
  </tbody>
</table>

<div class="text-center">
  <div class = "form-group row" style="color: aliceblue;">
    <label for="search" class="col-sm-2 col-form-label" style="text-align: right;">Search Date</label>
    <div class="col-sm-8">
      <input type="text" class="form-control" name="date" [(ngModel)]= "date"
         placeholder="Enter Match Date to search" >
    </div>
  </div>

  <div class = "form-group row">
    <div class="col-sm-2 offset-sm-7">
      <a class="btn btn-dark btn-lg active" role="button" (click)=
"searchMatch()" aria-pressed="true">Search Match</a>
    </div>
    <div class="col-sm-1">
      <a class="btn btn-light btn-lg active" role="button" (click)=
"clearSearch()" aria-pressed="true">Clear</a>
    </div>
  </div>
</div>

```

```

<div class = "form-group row">
    <div class="col-sm-2 offset-sm-7">
        <a class="btn btn-dark btn-lg active" role="button" (click)=
"generateMatch()" aria-pressed="true">Random Match</a>
    </div>
</div>
</div>

```

sort-match-list.component.ts

```

import { Component, OnInit } from '@angular/core';
import axios from 'axios';
import { OrderPipe } from 'ngx-order-pipe';

@Component({
  selector: 'app-sort-match-list',
  templateUrl: './sort-match-list.component.html',
  styleUrls: ['./sort-match-list.component.css']
})
export class SortMatchListComponent implements OnInit {

  matchArray = [];

  matches = [];
  order: string = 'date';
  date: string = '';

  constructor(private orderPipe: OrderPipe) {}

  ngOnInit(): void {
    axios.get("http://localhost:8080/matches").then(resp => {
      this.matches = resp.data;
      this.matchArray = this.orderPipe.transform(this.matches, this.order);
    });
    console.log(this.matchArray);
  }

  searchMatch(){
    if(this.date === ''){
      this.matchArray = this.orderPipe.transform(this.matches, this.order);
    }else{
      this.matchArray = this.matches.filter(match => {
        return match.date == this.date;
      })
    }
  }

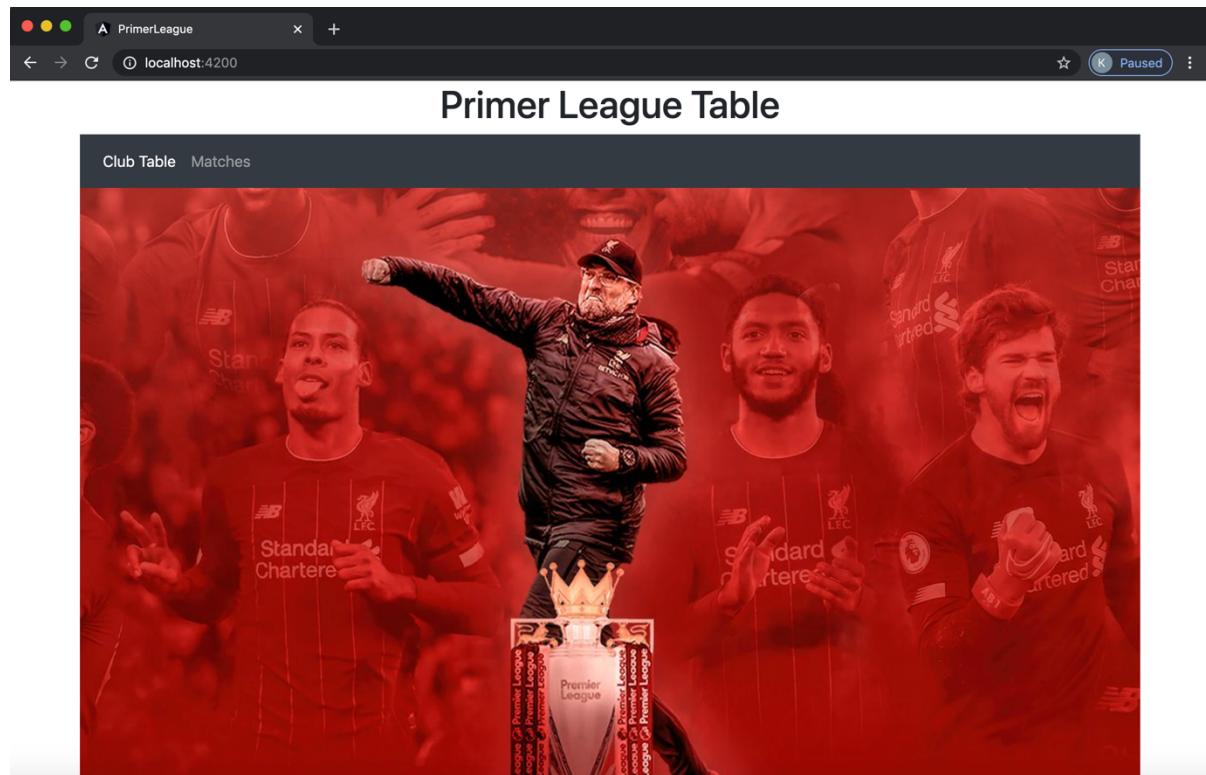
  clearSearch(){

```

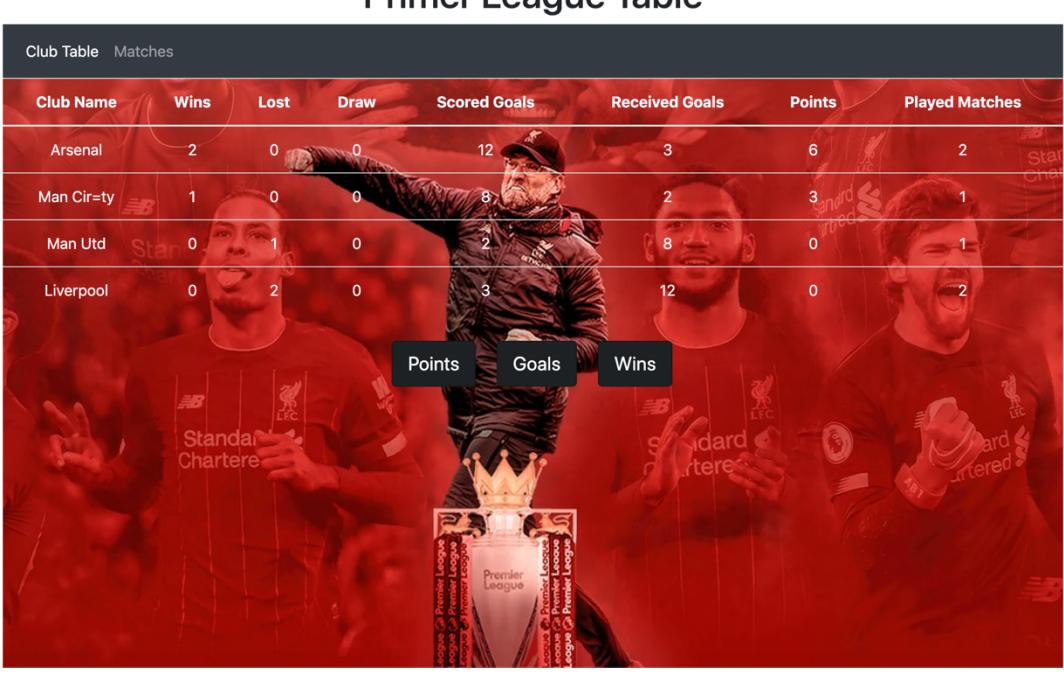
```
this.date = '';
this.searchMatch();
}

generateMatch(){
  axios.get("http://localhost:8080/generate").then(resp => {
    this.matches = resp.data;
    this.matchArray = this.orderPipe.transform(this.matches, this.order);
  });
}
}
```

Outputs

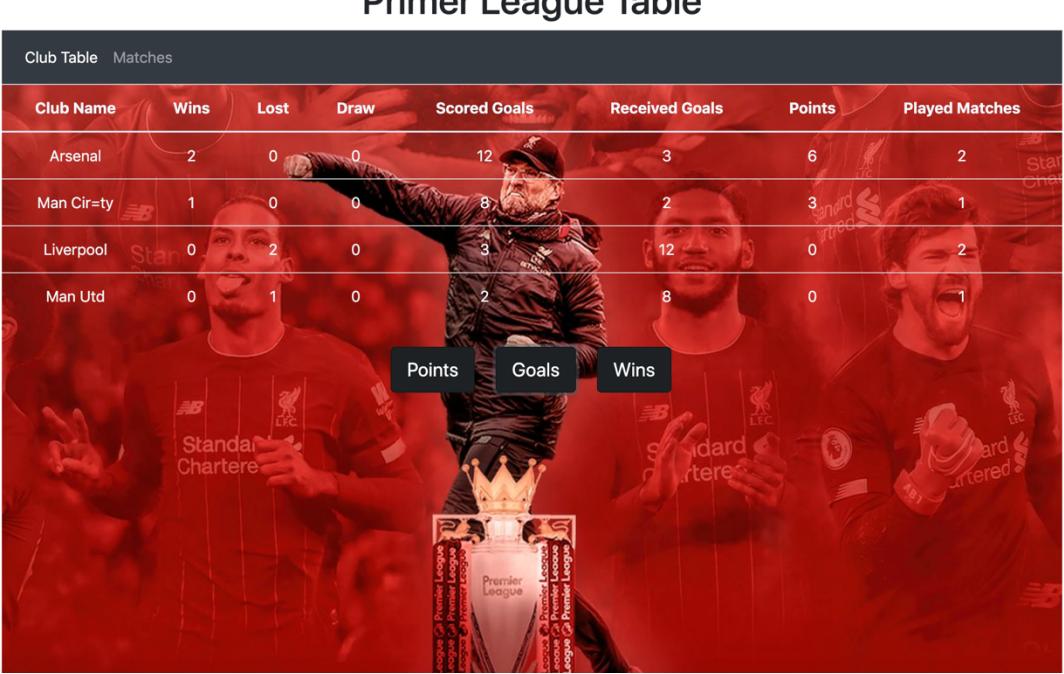


Points wise sorting



Club Name	Wins	Lost	Draw	Scored Goals	Received Goals	Points	Played Matches
Arsenal	2	0	0	12	3	6	2
Man Cir=ty	1	0	0	8	2	3	1
Man Utd	0	1	0	2	8	0	1
Liverpool	0	2	0	3	12	0	2

Scored goals wise sorting



Club Name	Wins	Lost	Draw	Scored Goals	Received Goals	Points	Played Matches
Arsenal	2	0	0	12	3	6	2
Man Cir=ty	1	0	0	8	2	3	1
Liverpool	0	2	0	3	12	0	2
Man Utd	0	1	0	2	8	0	1

Wins wise sorting

Primer League Table

Club Name	Wins	Lost	Draw	Scored Goals	Received Goals	Points	Played Matches
Arsenal	2	0	0	12	3	6	2
Man Cir=ty	1	0	0	8	2	3	1
Man Utd	0	1	0	2	8	0	1
Liverpool	0	2	0	3	12	0	2

Points Goals Wins

Primer League Table

Club For	Score For	Club Against	Score Against	Date
Arsenal	1	Liverpool	3	01-01-2021
Man Utd	2	Man Cir=ty	8	01-02-2021
Arsenal	9	Liverpool	2	01-03-2021

Search Date

Random Match

Search Match Clear

A PrimerLeague

localhost:4200/matches

Primer League Table

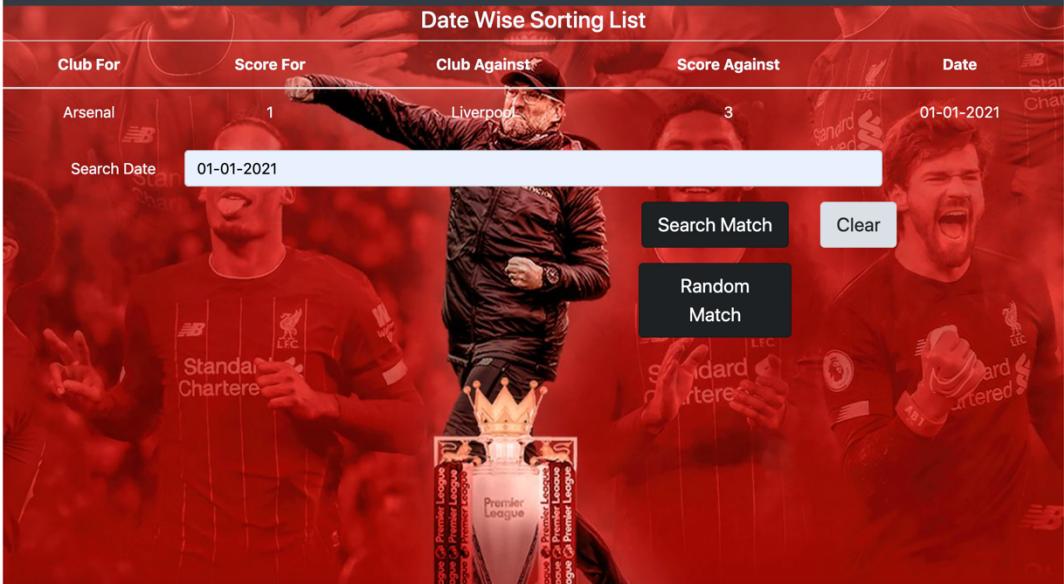
Club Table Matches

Date Wise Sorting List

Club For	Score For	Club Against	Score Against	Date
Arsenal	1	Liverpool	3	01-01-2021

Search Date: 01-01-2021

Search Match Clear Random Match



Generate random played match

A PrimerLeague

localhost:4200/matches

Primer League Table

Club Table Matches

Date Wise Sorting List

Club For	Score For	Club Against	Score Against	Date
Arsenal	1	Liverpool	3	01-01-2021
Man Utd	2	Man City	8	01-02-2021
Arsenal	9	Liverpool	2	01-03-2021
Liverpool	1	Man Utd	3	01-05-2021
Arsenal	6	Man City	1	01-06-2021

Search Date: Enter Match Date to search

Search Match Clear Random Match

