Report

Name : J.Thuwarakan

UoW ID : W1790265

IIT ID : 2019795

Title : OOP Course Work

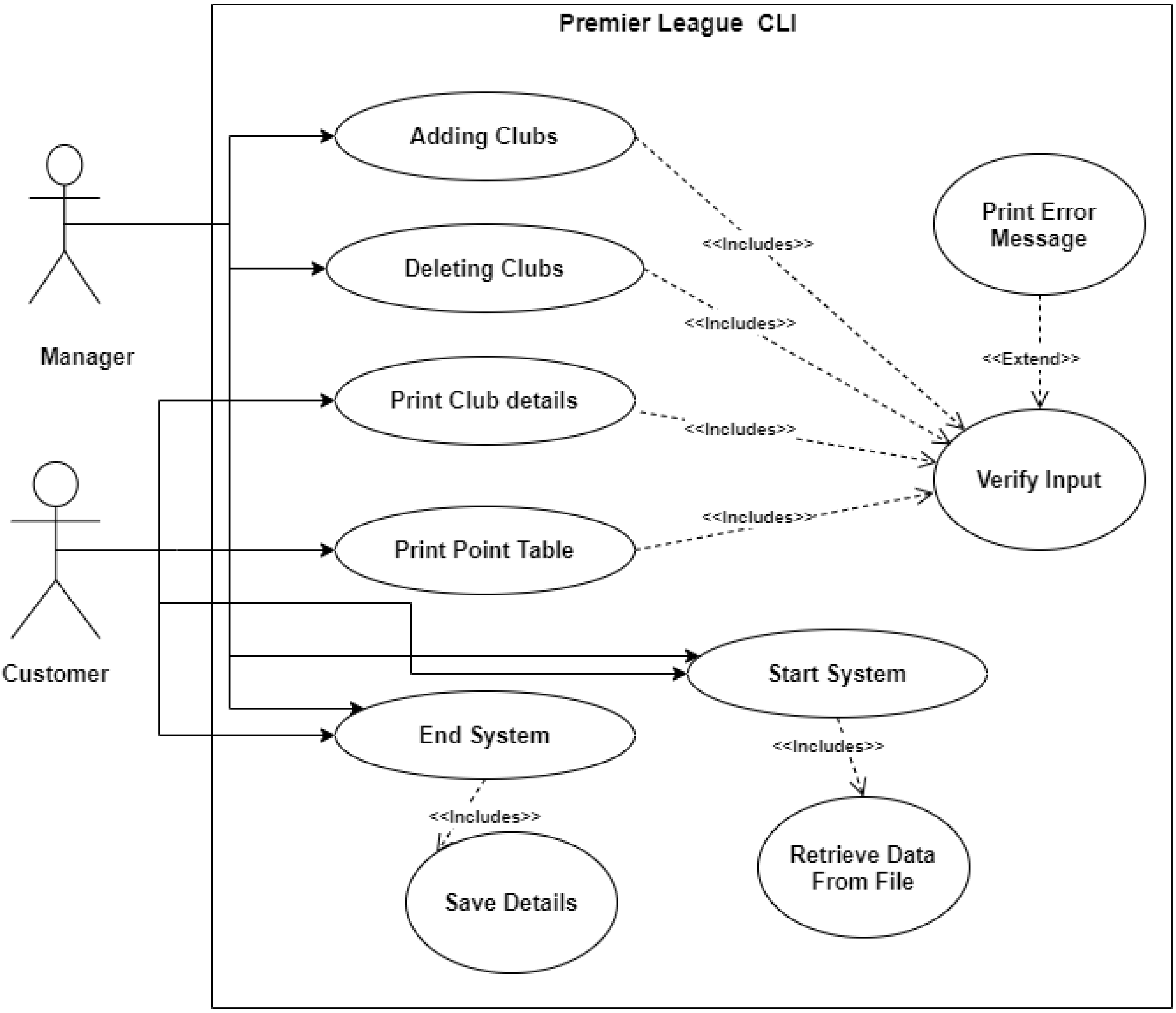
Date : 04/01/2021

"I confirm that I understand what plagiarism / collusion / contract cheating is and have read and understood the section on Assessment Offences in the Essential Information for Students. The work that I have submitted is entirely my own. Any work from other authors is duly referenced and acknowledged.

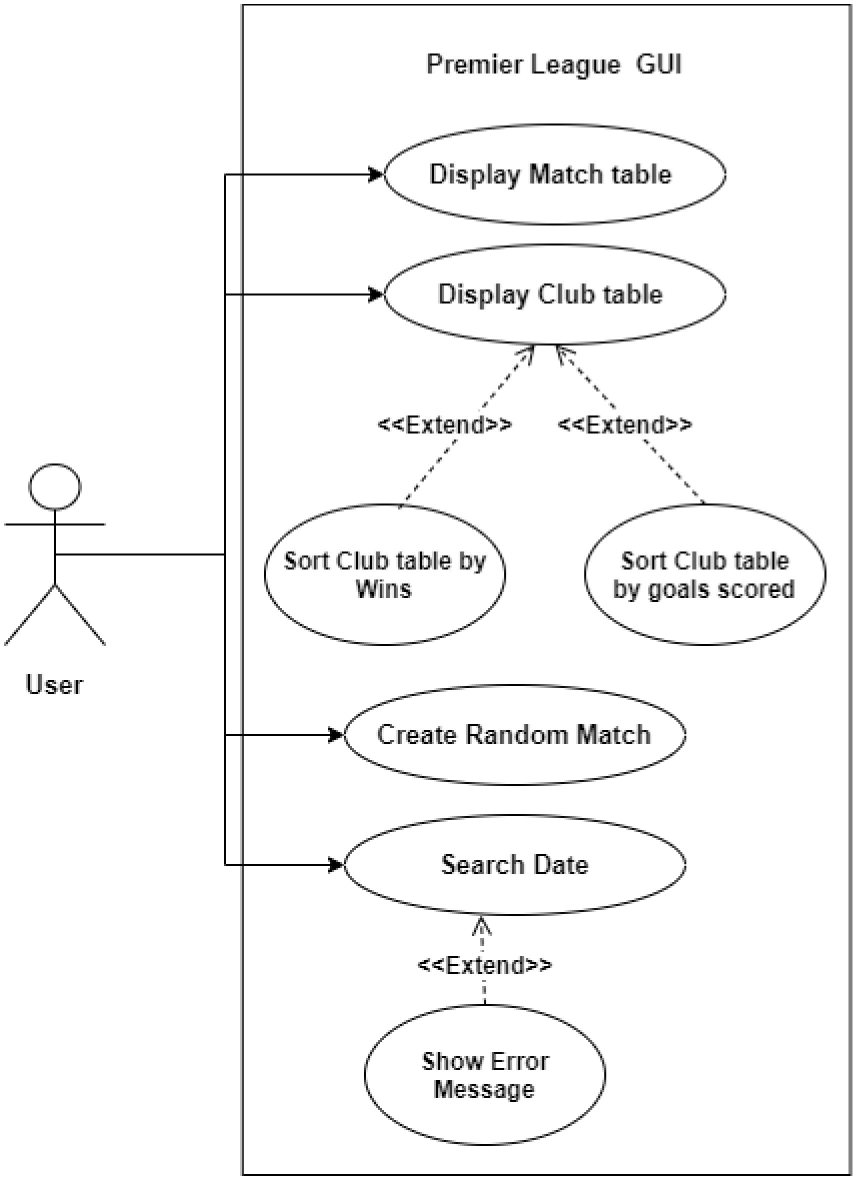
J.Thuwarakan
2019795

Project Design

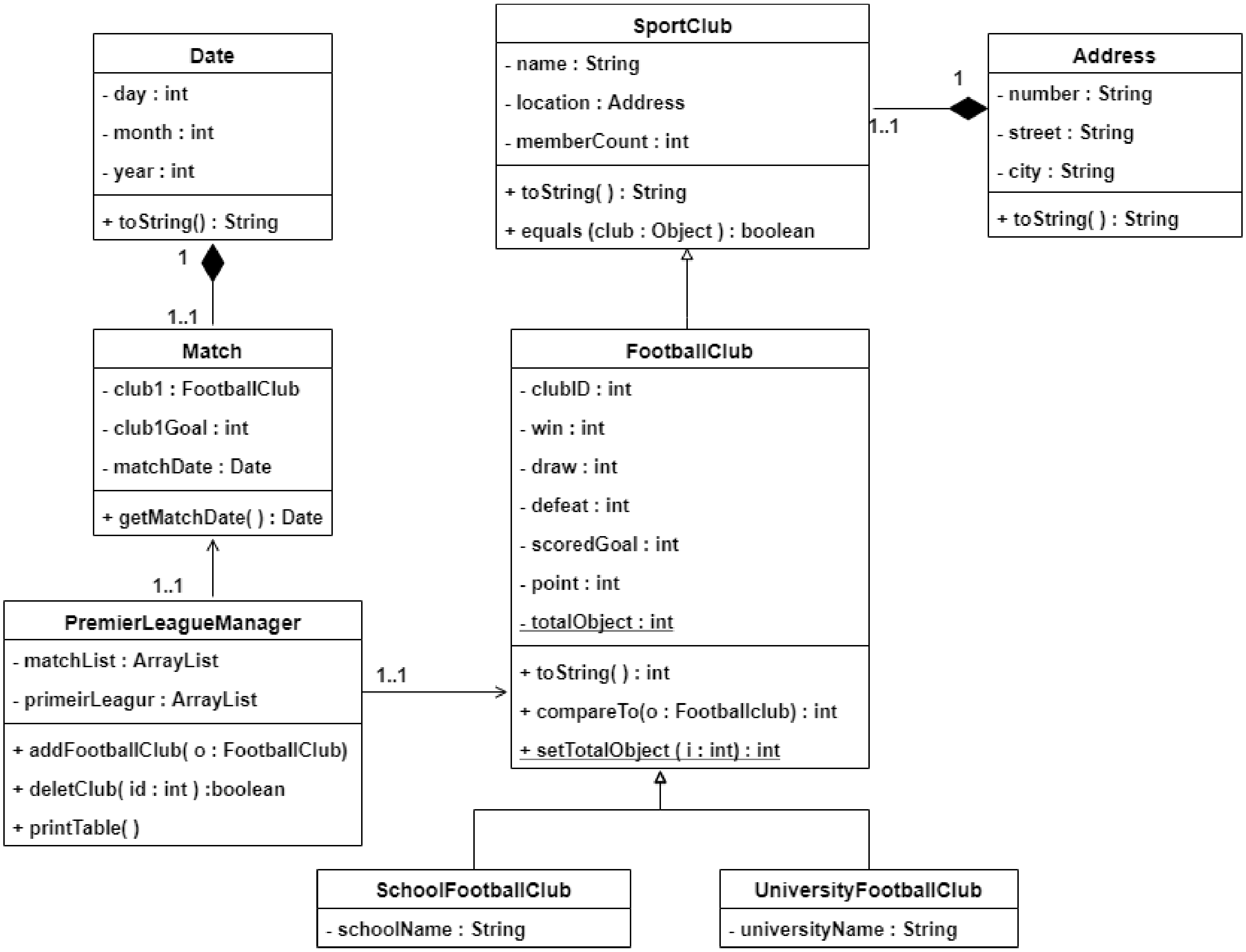
Use Case Diagram – 01 :-



Use Case diagram – 02 :-



Class diagram :-



Test plane for Java Course Work

TEST PLANE FOR JAVA COURSE WORK

Student name: J. Thuwarakan Student ID: 2019795

Test No	Test Inputs and instruction	Expected Result	Actual Result (State if not attempted)	Pass/Fail
1	Enter the method Number: 1 for select add Club method	Display "Add Method" and say select witch type of club you want to add	Display "Add Method" and say select witch type of club you want to add	pass
2	Enter the method Number: 2 For select Delete clubs	Display "Delete Method" and ask deleting club id	Display "Delete Method" and ask deleting club id	pass
3	Enter the method Number: 3 For select Print club method	Display "print club" method and ask print club id	Display "print club" method and ask print club id	pass
4	Enter the method Number: 4 For select print point table (if you add a member before)	Display Point table	Display Point table	pass
5	Enter the method Number: 4 For select print point table (if you didn't add a member before)	Display "Your total club count is 0 First add some clubs"	Display "Your total club count is 0 First add some clubs"	pass
6	Enter the method Number: 5 For add match method	Display the match method	Display the match method	pass
7	Enter the method Number: 6 For select Write file method (if you add a member before)	Save the member details in file	Save the member details in file	pass
8	Enter the method Number: 7 For save and end	Break the loop and Display "Club Data has been Successfully Saved to the Data File Match Data has been Successfully Saved to the Data File"	Break the loop and Display "Club Data has been Successfully Saved to the Data File Match Data has been Successfully Saved to the Data File"	pass
9	Enter an integer value above 7 or below 1 in the method number Ex: Enter the method Number: 8	Display "You entered a wrong number"	Display "You Enter a wrong number"	pass

	Enter the method Number: -1			
10	**Enter a string or a double value Ex: Enter method Number: thuwa Enter method Number: 1.1**	**Display “The System only accept 'Integer' Value”**	**Display “The System only accept 'Integer' Value”**	**pass**
11	**First select adding club method and select Number: 1 for adding Football club**	**Display the adding Football club method**	**Display the adding Football club method**	**pass**
12	**First select adding club method and select Number: 2 for adding University Football Club**	**Display the adding University Football Club method**	**Display the adding University Football Club method**	**Pass**
13	**First select adding club method and select Number: 3 for adding School Football Club**	**Display the adding School Football Club method**	**Display the adding School Football Club method**	**Pass**
14	**First select adding club method and Enter an integer value above 3 or below 1 in the method number Ex: Enter the method Number: 4 Enter the method Number: -1**	**Display” You Entered Invalid Number Please Enter Correct Number”**	**Display” You Entered Invalid Number Please Enter Correct Number”**	**pass**
15	**First Select adding club method then Enter a string or a double value Ex: Enter method Number: thuwa Enter method Number: 1.1**	**Display “Invalid input”**	**Display “Invalid input”**	**pass**
16	**First select adding club method and select the football club and enter the name and location then enter a string or double value for the member count**	**Display the error message and ask to re-enter the member count value**	**Display the error message and ask to re-enter the member count value**	**pass**
17	**First select adding club method and select the football club and enter the**	**Successfully add the football club and system display the club id**	**Successfully add the football club and system display the club id**	**pass**

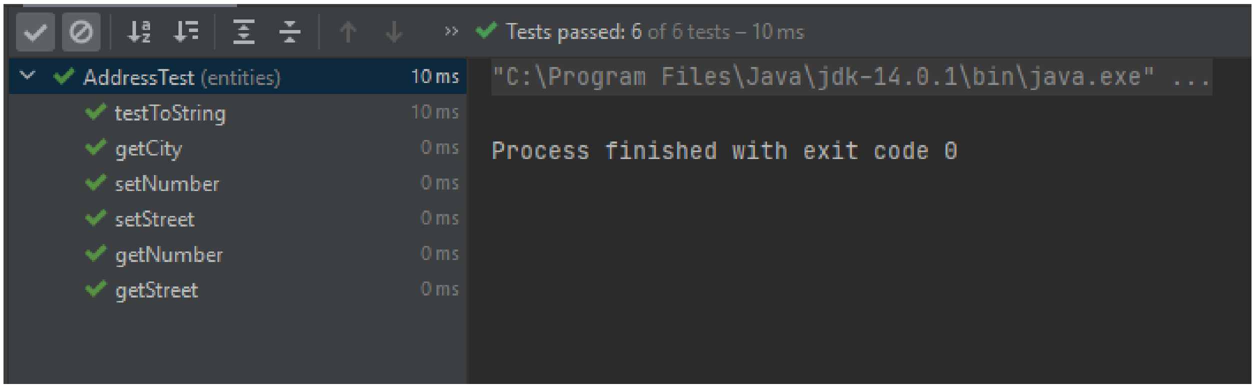
	name location and member count correctly			
18	First add a football club then go to delete club part and delete that club	System delete that club and display “you successfully deleted the club” message with that club name	System delete that club and display “you successfully deleted the club” message with that club name	pass
19	Go to the delete club part and enter a wrong club id	Display “You entered a wrong number”	Display “You entered a wrong number”	pass
20	Go to the delete club part and enter a string value	Display “That's not a number! Enter a Number for Delete Club :” and ask to enter the club id	Display “That's not a number! Enter a Number for Delete Club :” and ask to enter the club id	pass
21	First add one club then go to print club method and enter that club id	Display the club details	Display the club details	pass
20	Go to print club detail method and enter a wrong club id	System ask to enter the correct club id	System ask to enter the correct club id	pass
21	Go to print club detail method and enter the string or double value	System ask to enter the integer value	System ask to enter the integer value	pass
22	First add some clubs and Go to print point table method	Display Point table	Display Point table	pass
23	Do point table method without adding any clubs	Display “Your total club count is 0 First add some clubs”	Display “Your total club count is 0 First add some clubs”	pass
24	In the add match method enter a wrong id	System ask to enter the correct club id	System ask to enter the correct club id	pass
25	In the add match method enter a wrong goal count	System ask to enter the correct goal	System ask to enter the correct goal	pass
26	In the add match method enter a wrong date	System ask to enter the correct date	System ask to enter the correct date	pass

Unite Testing

Test method for Address class :-

```
public class AddressTest {
  Address location = new Address("12/b","Flower street","Colombo");

  @Test
  public void setNumber() {
    location.setNumber("13/b");
    assertEquals("13/b",location.getNumber());
  }

  @Test
  public void setStreet() {
    location.setStreet("Garden road");
    assertEquals("Garden road",location.getStreet());
  }


  @Test
  public void getNumber() {
    assertEquals("12/b",location.getNumber());
  }

  @Test
  public void getStreet() {
    assertEquals("Flower street",location.getStreet());
    }

    @Test
    public void getCity() {
        assertEquals("Colombo",location.getCity());
    }

    @Test
    public void testToString() {
        assertEquals("12/b, Flower street, Colombo.",location.toString());
    }
}
```

Result : -



Tests passed: 6 of 6 tests – 10 ms

```
AddressTest (entities)   10 ms
    testToString         10 ms
    getCity               0 ms
    setNumber             0 ms
    setStreet             0 ms
    getNumber             0 ms
    getStreet             0 ms
```

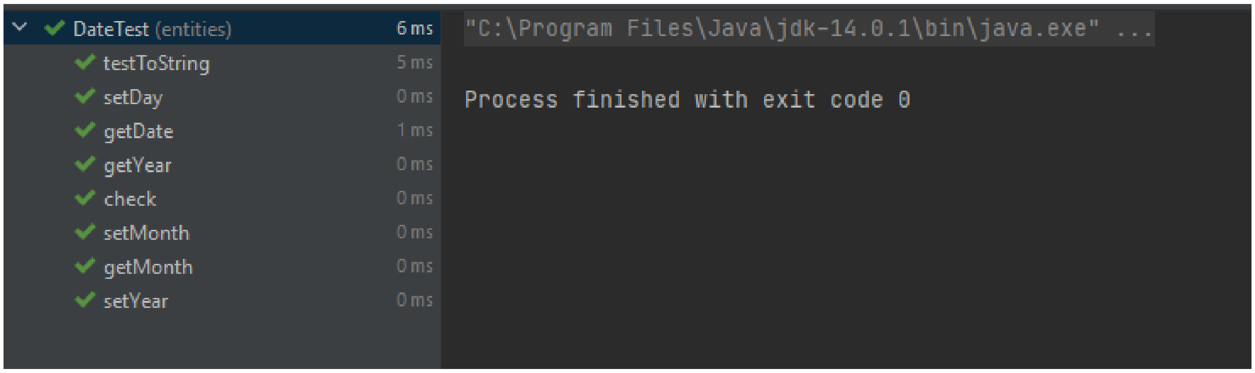
```
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" ...

Process finished with exit code 0
```

Test method for Date class :-

```
public class DateTest {
   Date date = new Date(12,4,2020);

   @Test
   public void setDay() {
      date.setDay(5);
      assertEquals(5,date.getDate());
   }

   @Test
   public void setMonth() {
      date.setMonth(8);
      assertEquals(8,date.getMonth());
   }

   @Test
   public void setYear() {
      date.setYear(2021);
      assertEquals(2021,date.getYear());
   }

   @Test
   public void getDate() {
      assertEquals(12,date.getDate());
   }

   @Test
   public void getMonth() {
      assertEquals(4,date.getMonth());
   }

   @Test
   public void getYear() {
      assertEquals(2020,date.getYear());
   }

   @Test
   public void testToString() {
      assertEquals("12/04/2020",date.toString());
```

```
        }

        @Test
        public void check() {
            Date date1 = new Date(12,4,2020);
            assertTrue(date.check(date1));
        }
    }
```

Result :-



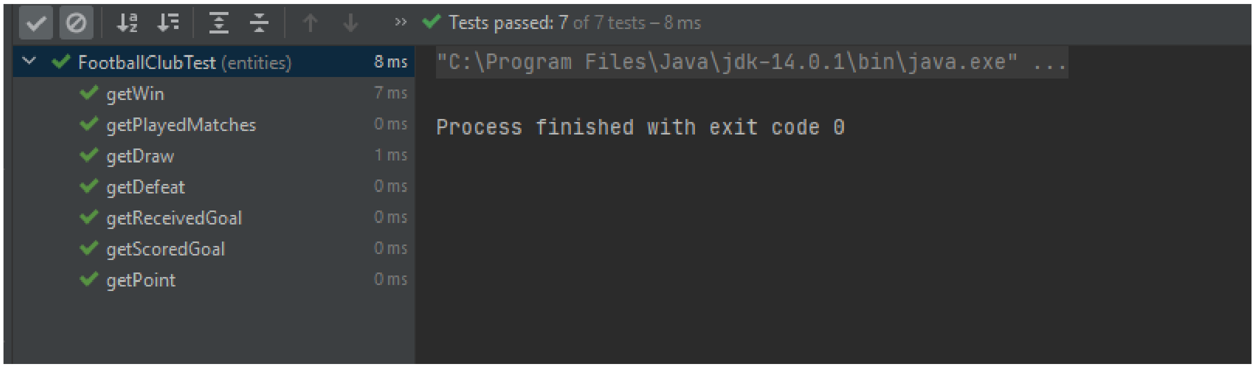
```
✓ DateTest (entities)      6 ms
    ✓ testToString         5 ms
    ✓ setDay               0 ms
    ✓ getDate              1 ms
    ✓ getYear              0 ms
    ✓ check                0 ms
    ✓ setMonth             0 ms
    ✓ getMonth             0 ms
    ✓ setYear              0 ms
```

```
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" ...

Process finished with exit code 0
```

Testing method for football club :-

```
public class FootballClubTest {
  FootballClub fb = new FootballClub(12,1,0,23,23,
      1,12 ,new Address("12","sdgsd","fgd"),"sfsf",1200);

  @Test
  public void getWin() {
    assertEquals(12,fb.getWin());
  }

  @Test
  public void getDraw() {
    assertEquals(1,fb.getDraw());
  }

  @Test
  public void getDefeat() {
    assertEquals(0,fb.getDefeat());
  }

  @Test
  public void getReceivedGoal() {
    assertEquals(23,fb.getReceivedGoal());
  }

  @Test
  public void getScoredGoal() {
        assertEquals(23,fb.getScoredGoal());
    }

    @Test
    public void getPoint() {
        assertEquals(1,fb.getPoint());
    }

    @Test
    public void getPlayedMatches() {
        assertEquals(12,fb.getPlayedMatches());
}

}
```

Result :-



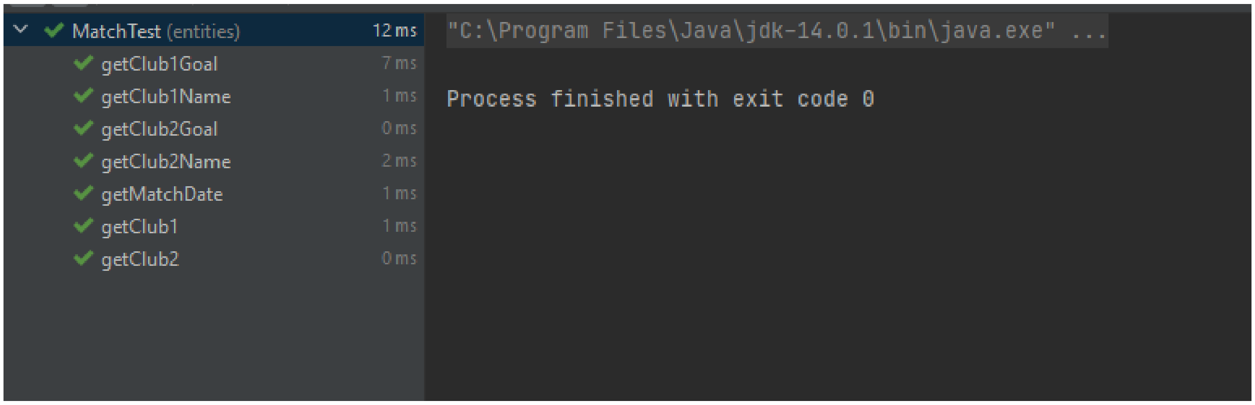
```
Tests passed: 7 of 7 tests – 8 ms

FootballClubTest (entities)   8 ms
    getWin                    7 ms
    getPlayedMatches          0 ms
    getDraw                   1 ms
    getDefeat                 0 ms
    getReceivedGoal           0 ms
    getScoredGoal             0 ms
    getPoint                  0 ms
```

```
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" ...

Process finished with exit code 0
```

Testing method for match class :-

```
public class MatchTest {
  Address loc = new Address("12","sdgfh","stweywj");
  FootballClub fb1 = new FootballClub(12,1,1,31,22,10,5,loc,"ghfgh",12);
  FootballClub fb2 = new FootballClub(13,1,1,41,32,0,5,loc,"ghfcgggh",120);
  Date date = new Date(12,11,2020);
  Match match = new Match(fb1,fb2,12,11,date,fb1.getName(), fb2.getName());

  @Test
  public void getClub1() {
    assertEquals(fb1,match.getClub1());
  }

  @Test
  public void getClub2() {
    assertEquals(fb2,match.getClub2());
  }

  @Test
  public void getClub1Goal() {
    assertEquals(12,match.getClub1Goal());
  }

  @Test
  public void getClub2Goal() {
    assertEquals(11,match.getClub2Goal());
  }

  @Test
  public void getMatchDate() {
    assertEquals(date,match.getMatchDate());
  }

  @Test
  public void getClub1Name() {
    assertEquals("ghfgh",match.getClub1Name());
  }

  @Test
  public void getClub2Name() {
    assertEquals("ghfcgggh",match.getClub2Name());
  }

}
```

Result :-



```
✓ MatchTest (entities)   12 ms
    ✓ getClub1Goal        7 ms
    ✓ getClub1Name        1 ms
    ✓ getClub2Goal        0 ms
    ✓ getClub2Name        2 ms
    ✓ getMatchDate        1 ms
    ✓ getClub1            1 ms
    ✓ getClub2            0 ms
```

```
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" ...

Process finished with exit code 0
```

Source code of CLI part

Sport club class code :-

```
// because we don't create objects for this class
public abstract class SportClub implements Serializable {
  private String name;
  private Address location;
  private int totalMemberCount;

  public SportClub(String name, Address location, int count){
    this.name = name;
    this.location = location;
    this.totalMemberCount = count;

  }

  // setter method of this class
  public void setName(String name){
    this.name = name;

  }

  public void setLocation(Address location){
    this.location = location;

  }

  public void setTotalMemberCount(int count){
    this.totalMemberCount = count;

  }

  // getter method of this class
  public String getName(){
    return name;

  }

  public Address getLocation(){
    return location;

  }

  public int getTotalMemberCount(){
    return totalMemberCount;
    }

    @Override
    public String toString(){
        return "Name of the club : " + name + "\nAddress : " + location + "\nTotal Member Count : "
+ totalMemberCount;

    }

    // this override methods help to sort the sport club
    @Override
    public boolean equals(Object club) {
        if (this == club) return true;
        if (club == null || getClass() != club.getClass()) return false;
        SportClub = (SportClub) club;
        return name.equals(sportClub.name) ;

    }

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

}
```

Football club code :-

```
public class FootballClub extends SportClub implements Serializable, Comparable<FootballClub>
{
  // we use club id to identify each object and  it's unique for each objects.
  private int clubID;
  private int win;
  private int draw;
  private int defeat;
  private int receivedGoal;
  private int scoredGoal;
  // point also want to be a int value because it can't be a decimal value
  private int point;
  private int playedMatches;
  //this static attribute is store the number of total created numbers
  private static int totalObjects;


  public FootballClub(int win, int draw, int defeat, int receivedGoal, int scoredGoal, int point, int
playedMatches, Address location, String name, int count){
    super(name,location,count);
    totalObjects = totalObjects + 1;
    this.win = win;
    this.draw = draw;
    this.defeat = defeat;
    this.receivedGoal = receivedGoal;
    this.scoredGoal = scoredGoal;
    this.point = point;
    this.playedMatches = playedMatches;
    this.clubID = 1000 + totalObjects;

  }

  //setter methods
  //clubID will automatically generate
  public void setWin(int win) {
    this.win = win;

  }

  public void setDraw(int draw) {
    this.draw = draw;

  }

  public void setDefeat(int defeat) {
    this.defeat = defeat;
```

```
}

public void setReceivedGoal(int receivedGoal) {
    this.receivedGoal = receivedGoal;

}

public void setScoredGoal(int scoredGoal) {
    this.scoredGoal = scoredGoal;

}

public void setPoint(int point) {
    this.point = point;

}

public void setPlayedMatches(int playedMatches) {
    this.playedMatches = playedMatches;

}

public static void setTotalObjects(int totalObjects) {
    FootballClub.totalObjects = totalObjects;
}

// getter method
public int getClubID() {
    return clubID;

}

public int getWin() {
    return win;

}

public int getDraw() {
    return draw;

}

public int getDefeat() {
    return defeat;

}

public int getReceivedGoal() {
        return receivedGoal;

    }

    public int getScoredGoal() {
        return scoredGoal;

    }

    public int getPoint() {
        return point;

    }

    public int getPlayedMatches() {
        return playedMatches;

    }

    @Override
    public String toString() {
        String details;
        details = super.toString()+"\nClub ID : "+ clubID + "\nPlayed Matches : " +playedMatches
+"\nTotal Win : "+ win + "\nTotal Defeat : " + defeat + "\nTotal Draw : " + draw
                +"\nReceived Goal : " + receivedGoal + "\nScored Goal : " + scoredGoal + "\nTotal Point
: " + point;
        return details;

    }


    @Override
    public int compareTo(FootballClub club) {
        int thisScore = scoredGoal - receivedGoal;
        int clubScore = club.getScoredGoal() - club.getReceivedGoal();
        if(point == club.getPoint()){
            if(thisScore > clubScore){
                return -1;
            }
            else if(thisScore < clubScore){
                return 1;
            }
            else{
                return 0 ;
            }
        }
        else if (point > club.getPoint()){
            return -1;
        }
        else{
          return 1;
        }
    }
}
```

University Football club code :-

```
  public class UniversityFootballClub extends FootballClub {
        private String universityName;


        public UniversityFootballClub(int win, int draw, int defeat, int receivedGoal, int scoredGoal,
int point,
                                    int playedMatches, Address location, String name, int count, String
universityName) {
            super(win, draw, defeat, receivedGoal, scoredGoal, point, playedMatches, location,
name, count);
            this.universityName = universityName;
        }

        public String getUniversityName() {
            return universityName;

        }

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

        }

        @Override
        public String toString() {
            return  super.toString()+ "\nschoolName= " + universityName;

        }

}
```

School football club code :-

```
public class UniversityFootballClub extends FootballClub {
    private String universityName;


    public UniversityFootballClub(int win, int draw, int defeat, int receivedGoal, int scoredGoal,
int point, int playedMatches, Address location, String name, int count, String universityName) {
      super(win, draw, defeat, receivedGoal, scoredGoal, point, playedMatches, location,
name, count);
      this.universityName = universityName;
    }

    public String getUniversityName() {
      return universityName;

    }

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

    }

    @Override
    public String toString() {
      return  super.toString()+ "\nschoolName= " + universityName;

    }

}
```

Address class code

```
public class Address implements Serializable {
  // number can be 1B or 1/A so i make it a string variable
  private String number;
  private String street;
  private String city;

  public Address(String no, String street, String city) {
    this.number = no;
    this.street = street;
    this.city = city;

  }

  // setter methods
  public void setNumber(String no) {
    this.number = no;

  }

  public void setStreet(String street) {
    this.street = street;

  }

  public void setCity(String city) {
    this.city = city;

  }

  // getter methods
  public String getNumber() {
    return number;

  }

  public String getStreet() {
    return street;

  }

  public String getCity() {
    return city;

  }

  @Override
```

```
   public String toString() {
      return number + ", " + street + ", " + city + ".";

   }
}
```

Date class code :-

```
public class Date implements Serializable {
   private int day;
   private int month;
   private int year;

   public Date(int day, int month, int year){
      this.day = day;
      this.month = month;
      this.year = year;

   }

   //setter methods
   public void setDay(int day){
      this.day = day;

   }

   public void setMonth(int month){
      this.month = month;

   }

   public void setYear(int year){
      this.year = year;

   }

   //getter method
   public int getDate(){
      return day;

   }

   public int getMonth(){
      return month;

   }
```

```
    public int getYear(){
        return year;

    }

    @Override
    public String toString() {
        return String.format("%02d/%02d/%s",day,month,year);

    }

    public boolean check(Date date){
        return day == date.getDate() && month == date.getMonth() && year == date.getYear();
    }

}
```

Match class code :-

```
public class Match implements Serializable{
    private FootballClub club1;
    private FootballClub club2;
    private int club1Goal;
    private int club2Goal;
    private Date matchDate;
    private String club1Name;
    private String club2Name;
    private String stringDate;

    public Match(FootballClub club1, FootballClub club2, int club1Goal, int club2Goal, Date date, String name1, String name2) {
        this.club1 = club1;
        this.club2 = club2;
        this.club1Goal = club1Goal;
        this.club2Goal = club2Goal;
        this.matchDate = date;
        this.club1Name = name1;
        this.club2Name = name2;
        this.stringDate = date.toString();
    }

    public void setClub1(FootballClub club1) {
        this.club1 = club1;

    }

    public void setClub2(FootballClub club2) {
```

```
    this.club2 = club2;

}

public void setClub1Goal(int club1Goal) {
    this.club1Goal = club1Goal;

}

public void setClub2Goal(int club2Goal) {
    this.club2Goal = club2Goal;

}

public FootballClub getClub1() {
    return club1;

}

public FootballClub getClub2() {
    return club2;

}

public int getClub1Goal() {
    return club1Goal;

}

public int getClub2Goal() {
    return club2Goal;

}

public Date getMatchDate() {
    return matchDate;
}

public void setMatchDate(Date matchDate) {
    this.matchDate = matchDate;
}

public String getClub1Name() {
    return club1Name;
}

public void setClub1Name(String club1Name) {
    this.club1Name = club1Name;
    }

    public String getClub2Name() {
        return club2Name;
    }

    public void setClub2Name(String club2Name) {
        this.club2Name = club2Name;
    }

    public String getStringDate() {
        return stringDate;
    }

    public void setStringDate(String stringDate) {
        this.stringDate = stringDate;
    }
}
```

Code of league manager interface :-

```
public interface LeagueManager {

        public void addClub(FootballClub club);
        public boolean deleteClub(int clubID);
        public boolean printClubDetails(int clubID);
        public void printTable();
        public ArrayList<Object> getClub(int id);
        public void addMatch(Match match);
        public void addFile(String fileName) throws IOException;
        public void loadData(String fileName) throws IOException, ClassNotFoundException;

    }
```

Premier league manager code

```
public class PremierLeagueManager implements LeagueManager{
    private final List<FootballClub> premierLeague = new ArrayList<>();
    private final List<Match> matchList = new ArrayList<>();
    private final int totalTeams;

    public PremierLeagueManager (int count){
        this.totalTeams = count;
    }
    //add method
    public void addClub(FootballClub club){
        // check this club is already in the list
        for(SportClub league : premierLeague) {
            if (club.equals(league)) {
                System.out.println("This club already in the premier league\nplease add another
club");
                return;
            }
        }
        if(premierLeague.size() < totalTeams){
            premierLeague.add(club);
            System.out.println("The club is Successfully added and The ID : "+club.getClubID());
        }
        else{
            System.out.println("The slot is full");
        }
    }
    //---------------------------------------------------------------------------------------
    //deleting club
    public boolean deleteClub(int clubID){
        boolean flag = false;
        if(premierLeague.size() == 0){
            System.out.println("\nYour total Club count is 0 \nFirst add some Club");
            flag = true;
        }
        else{
            for(FootballClub club : premierLeague) {
                if (club.getClubID() == clubID) {
                    premierLeague.remove(club);
                    System.out.println("Football Club " + club.getName() + " is Successfully Deleted");
                    flag = true;
                    break;
                }
            }
        }
        return flag;
    }
```

```
  //------------------------------------------------------------------------------------
  //display details of particular club
  public boolean printClubDetails(int clubID){
    boolean flag = false;
    if(premierLeague.size() == 0) {
      System.out.println("\nYour total Club count is 0 \nFirst add some Club");
      flag = true;
    }
    else{
      for (FootballClub Club : premierLeague) {
        int check = Club.getClubID();
        if (clubID == check) {
          System.out.println("\n"+ Club + "\n");
          flag = true;
          break;
        }
      }
    }
    return flag;
  }
  //-------------------------------------------------------------------------------------
  //print point table
  public void printTable() {
    if (premierLeague.size() == 0) {
      System.out.println("\nYour total Club count is 0 \nFirst add some Club");
    } else {
      System.out.println("                    Premier League Point Table\n");
      System.out.println("+------------------------+----------------+--------------------+---------------+----
--------------+");
      System.out.printf("%s %20s %5s %10s  %5s %15s %5s %10s %5s %15s %3s%n", "|", "Club
Name", "|", "Club Id", "|","Played Matches","|" ,"Point", "|", "Goal Different", "|");
      System.out.println("+------------------------+----------------+--------------------+---------------+----
--------------+");
      Collections.sort(premierLeague);
      for (FootballClub footballClub : premierLeague) {
        System.out.printf("%s %20s %5s %10s  %5s %15s %5s %10s %5s %15s %3s%n", "|",
footballClub.getName(), "|", footballClub.getClubID(),
            "|", footballClub.getPlayedMatches(), "|", footballClub.getPoint(), "|",
(footballClub.getScoredGoal() - footballClub.getReceivedGoal()), "|");
        //System.out.println("+-------------------+---------------+-------------------+---------------+---
---------------+");
      }
      System.out.println("+------------------------+----------------+--------------------+---------------+----
--------------+");
    }
  }
  //-------------------------------------------------------------------------------------------
  //getClub method
```

```
public  ArrayList<Object> getClub(int id){
    boolean flag = false;
    FootballClub club = null;
    for (FootballClub footballClub : premierLeague) {
        int check = footballClub.getClubID();
        if (check == id) {
            club = footballClub;
            flag = true;
            break;
        }
    }
    return new ArrayList<Object>(  Arrays.asList(club,flag));
}
//------------------------------------------------------------------------------------------
//addMatch method
public void addMatch(Match match){
    matchList.add(match);

    int club1Goal = match.getClub1Goal();
    int club2Goal = match.getClub2Goal();
    FootballClub club1 = match.getClub1();
    FootballClub club2 = match.getClub2();
    //set received and scored goals
    club1.setScoredGoal(club1.getScoredGoal() + club1Goal);
    club1.setReceivedGoal(club1.getReceivedGoal() + club2Goal);
    club2.setScoredGoal(club2.getScoredGoal() + club2Goal);
    club2.setReceivedGoal(club2.getReceivedGoal() + club1Goal);
    //update matches
    club1.setPlayedMatches(club1.getPlayedMatches() + 1);
    club2.setPlayedMatches(club2.getPlayedMatches() + 1);

    //check which club is win and update point,win,defeat,draw
    if(club1Goal > club2Goal){
        club1.setPoint(club1.getPoint() + 2);
        club1.setWin(club1.getWin() + 1);
        club2.setDefeat(club2.getDefeat() + 1);

    }
    else if(club1Goal < club2Goal){
        club2.setPoint(club2.getPoint() + 2);
        club2.setWin(club2.getWin() + 1);
        club1.setDefeat(club1.getDefeat() + 1);

    }
    else{
        club1.setPoint(club1.getPoint() + 1);
        club2.setPoint(club2.getPoint() + 1);
        club1.setDraw(club1.getDraw() + 1);
    club2.setDraw(club2.getDraw() + 1);

  }

}
//-----------------------------------------------------------------------------------------------------------
//addFile method
public void addFile(String fileName) throws IOException {

  FileOutputStream fileOutputStream = new FileOutputStream(fileName);
  ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);

  for (FootballClub footballClub : premierLeague) {
    objectOutputStream.writeObject(footballClub);
  }

  objectOutputStream.flush();
  fileOutputStream.close();
  objectOutputStream.close();

  System.out.println("\nClub Data has been Successfully Saved to the Data File");
}
//-----------------------------------------------------------------------------------------------------------
//getData method
public void loadData(String fileName) throws IOException, ClassNotFoundException {
  FileInputStream fileInputStream = new FileInputStream(fileName);
  ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

  for(;;) {
    try {
      FootballClub club = (FootballClub) objectInputStream.readObject();
      premierLeague.add(club);

    }
    catch (EOFException e) {
      break;
    }
  }
   int clubID = 1000;
  for(FootballClub club : premierLeague){
    if(clubID < club.getClubID()){
      clubID = club.getClubID();
    }
  }

  int count = clubID - 1000;
  FootballClub.setTotalObjects(count);
```

```
    fileInputStream.close();
    objectInputStream.close();

    System.out.println("\nClub Data has been Successfully Loaded from the File");
  }
  //-------------------------------------------------------------------------------------------------------------------------------------
  //add file method
  public void addFileMatch(String fileName) throws IOException {

    FileOutputStream fileOutputStream = new FileOutputStream(fileName);
    ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);

    for (Match match : matchList) {
      objectOutputStream.writeObject(match);
    }

    objectOutputStream.flush();
    fileOutputStream.close();
    objectOutputStream.close();

    System.out.println("\nMatch Data has been Successfully Saved to the Data File");
  }
  //-------------------------------------------------------------------------------------------------------------------------------------
  //getData method
  public void loadMatchData(String fileName) throws IOException, ClassNotFoundException {
    FileInputStream fileInputStream = new FileInputStream(fileName);
    ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

    for(;;) {
      try {
        Match match = (Match) objectInputStream.readObject();
        matchList.add(match);
      }
      catch (EOFException e) {
        break;
      }
    }
    fileInputStream.close();
    objectInputStream.close();
    System.out.println("\nMatch Data has been Successfully Loaded from the File");
  }
  //-------------------------------------------------------------------------------------------------------------------
}
```

Console menu code :-

```
public class ConsoleMenu {
  private static final PremierLeagueManager manager = new PremierLeagueManager(20);

  public static void main(String[] args) throws IOException, ClassNotFoundException {
    try {
      manager.loadData("Data.txt");
      manager.loadMatchData("Match.txt");
    } catch (EOFException e) {
      System.out.println("FIle is empty");
    }catch(ArrayIndexOutOfBoundsException ae){
      System.out.println("File is empty");
    }
    //Application.launch(args);
    menu:
    while (true) {
      System.out.println("\n================== Welcome to the Premier League
Management System ==================");
      System.out.println("\nYou can select which method you Want by Entering Number\n ");
      System.out.println("1 = Adding Clubs to Premier League");
      System.out.println("2 = Delete Club for Premier League");
      System.out.println("3 = print Club Details ");
      System.out.println("4 = print Point Table of Premier League");
      System.out.println("5 = add Match to Premier League");
      System.out.println("6 = Save and End the Programme");

      System.out.print("\nEnter the Method Number : ");
      try {
        Scanner input = new Scanner(System.in);
        int count = input.nextInt();

        switch (count) {
          case 1:
            FootballClub club;
            System.out.println("Enter which type of club you want to add ?");
            System.out.println("1 = Football Club");
            System.out.println("2 = University Football Club");
            System.out.println("3 = School Football Club");
            System.out.print("Enter : ");
            int select = input.nextInt();

            if (select > 0 && select < 4) {

              System.out.print("Enter club Name : ");
              input.nextLine();
              String Name = input.nextLine();
              System.out.println("Now enter the address of club");
              System.out.print("Enter the No : ");

              String no = input.nextLine();
              System.out.print("Enter the Street Name : ");

              String street = input.nextLine();
              System.out.print("Enter the City Name : ");
              String city = input.nextLine();

              int memberCount = checkInput("Enter the Member count of this club : ");

              Address address = new Address(no, street, city);

              switch (select) {
                case 1:
                  club = new FootballClub(0, 0, 0, 0, 0,
                      0, 0, address, Name, memberCount);
                  manager.addClub(club);
                  break;
                case 2:
                  System.out.print("Enter University Name : ");
                  input.nextLine();
                  String uniName = input.nextLine();
                  club = new FootballClub(3, 3, 3, 3, 3,
                      3, 3, address, Name, memberCount);
                  manager.addClub(club);
                  break;
                case 3:
                  System.out.print("Enter School Name : ");
                  input.nextLine();
                  String schoolName = input.nextLine();
                  club = new FootballClub(0, 0, 0, 0, 0,
                      0, 0, address, Name, memberCount);
                  manager.addClub(club);
                  break;
              }
            } else {
              System.out.println("\nYou Entered Invalid Number Please Enter Correct
Number.");
            }
            break;
          //-----------------------------------------------------------------------------------------------
          case 2:
            int clubID;
            boolean flag ;
            do {
              System.out.print("Enter the club ID for Delete Club : ");
```

```
        while (!input.hasNextInt()) {
            System.out.print("That's not a number! Enter a Number for Delete Club : ");
            input.next();
        }
        clubID = input.nextInt();
        flag = manager.deleteClub(clubID);
        if (!flag) {
            System.out.println("Invalid Id Enter the correct ID");
        }

    } while (!flag);
    break;
//------------------------------------------------------------------------------------------
case 3:
    int details;
    boolean flag2 ;
    do {
        System.out.print("Enter the club ID for print details : ");
        while (!input.hasNextInt()) {
            System.out.print("That's not a number! Enter a Number for print Details : ");
            input.next();
        }
        details = input.nextInt();
        flag2 =manager.printClubDetails(details);
        if (!flag2) {
            System.out.println("Invalid Id Enter the correct ID");
        }
    } while (!flag2);
    break;
//------------------------------------------------------------------------------------------
case 4:
    manager.printTable();
    break;
//------------------------------------------------------------------------------------------
case 5:
    // get football clubs to add in a match
    FootballClub matchClub1 = checkID("First");
    FootballClub matchClub2 = checkID("Second");
    //--------------------------------------------------------------
    // check and get the goals of two teams
    int goal1 = checkInput("Enter the first Team Goals : ") ;
    int goal2 = checkInput("Enter the second5 Team Goals : ");

    int year;
    do {
        System.out.print("Enter the Year : ");
        while (!input.hasNextInt()) {
            System.out.print("That's not a Year! Enter a correct year : ");
```

```
                input.next();
              }
              year = input.nextInt();
              if (year < 2019) {
                System.out.println("You entered a wrong year ");
              }
            } while (year < 2019);
            //------------------------------------------------
            int month;
            do {
              System.out.print("Enter the month : ");
              while (!input.hasNextInt()) {
                System.out.print("Enter the month in Number format : ");
                input.next();
              }
              month = input.nextInt();
              if (month > 12 ||month < 0) {
                System.out.println("The month is should be in 1 to 12");
              }
            } while (month > 12 || month < 0);
            // ---------------------------------------------------
            int[] DaysInMonth = {31,28,31,30,31,30,31,31,30,31,30,31};
            int day;
            do {
              System.out.print("Enter the day : ");
              while (!input.hasNextInt()) {
                System.out.print("Enter the day in Number format : ");
                input.next();
              }
              day = input.nextInt();
              if (day >= DaysInMonth[month-1] || day<0) {
                System.out.println("The day is should be in 1 to " + DaysInMonth[month-1]);
              }
            } while (day >= DaysInMonth[month-1] || day < 0);
            //------------------------------------------------------

            Date date = new Date(day, month, year);
            Match match = new Match(matchClub1, matchClub2, goal1, goal2,
date,matchClub1.getName(),matchClub2.getName());
            manager.addMatch(match);
            break;
          //-------------------------------------------------------------------------------

          //--------------------------------------------------------------------------------------
          case 6:
            manager.addFile("Data.txt");
            manager.addFileMatch("Match.txt");
            break menu;
            //------------------------------------------------------------------------------------------
            default:
               System.out.println("You entered a wrong number");
         }
      }
      catch(InputMismatchException e){
         System.out.println("The System only accept 'Integer' Value");
      }
   }
}
//----------------------------------------------------------------------------------
//check the id and return the football club
public static FootballClub checkID(String number){
   FootballClub matchClub;
   boolean flag ;
   int club;
   do {
      System.out.print("Enter the " + number + " Team ID : ");
      Scanner input = new Scanner(System.in);
      while (!input.hasNextInt()) {
         System.out.print("ID should be in number format! Enter a Number for Delete Club : ");
         input.next();
      }
      club = input.nextInt();
      ArrayList<Object> p = manager.getClub(club);
      flag = (boolean) p.get(1);
      matchClub = (FootballClub) p.get(0);

      if (!flag) {
         System.out.println("Invalid Id Enter the correct ID");
      }
   } while (!flag);
   return matchClub;
}
//---------------------------------------------------------------------------------
//check the integer inputs
public static int checkInput(String Question){
   int number;
   do {
      Scanner input = new Scanner(System.in);
      System.out.print(Question);
      while (!input.hasNextInt()) {
         System.out.print("That's not a number! Enter a Number value : ");
         input.next();
      }
      number = input.nextInt();
      if (number < 0) {
         System.out.println("Enter a positive value");
```

```
            }
        } while (number < 0);
        return number;
    }


}
```

Source code of backend

League Controller method code :-

```
public class LeagueController {

  public Result getClubs() throws IOException {
    ArrayList<FootballClub> result = LeagueService.getInstance().getClub();
    ObjectMapper map = new ObjectMapper();

    JsonNode data = map.convertValue(result,JsonNode.class);
    return ok(data);
  }

  public Result getMatches() throws IOException {
    ArrayList<Match> result = LeagueService.getInstance().getMatch();
    ObjectMapper map = new ObjectMapper();

    JsonNode data = map.convertValue(result,JsonNode.class);
    return ok(data);
  }

  public Result getClubsWin() throws IOException {
    ArrayList<FootballClub> result = LeagueService.getInstance().getWinClub();
    ObjectMapper map = new ObjectMapper();

    JsonNode data = map.convertValue(result,JsonNode.class);
    return ok(data);
  }

  public Result getClubsScore() throws IOException {
    ArrayList<FootballClub> result = LeagueService.getInstance().getScoreClub();
    ObjectMapper map = new ObjectMapper();

    JsonNode data = map.convertValue(result,JsonNode.class);
    return ok(data);
  }

}
```

League Service class code :-

```
public class LeagueService {
  private static LeagueService instance;

  public static LeagueService getInstance() {
    if (instance == null) {
      instance = new LeagueService();
    }
    return instance;
  }

  public ArrayList<FootballClub> getClub() throws IOException {
    ArrayList<FootballClub> clubs = new ArrayList<>();

    FileInputStream fileInputStream = new FileInputStream("Data.txt");
    ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

    for (; ; ) {
      try {
        FootballClub club = (FootballClub) objectInputStream.readObject();
        clubs.add(club);
      } catch (EOFException | ClassNotFoundException e) {
        break;
      }
    }
    fileInputStream.close();
    objectInputStream.close();
    Collections.sort(clubs);

    return clubs;

  }

  public ArrayList<Match> getMatch() throws IOException {
    ArrayList<Match> matches = new ArrayList<>();

    FileInputStream fileInputStream = new FileInputStream("Match.txt");
    ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

    for (; ; ) {
      try {
        Match match = (Match) objectInputStream.readObject();
        matches.add(match);
      } catch (EOFException | ClassNotFoundException e) {
        break;
      }
    }
```

```
        fileInputStream.close();
        objectInputStream.close();

        return matches;
    }

    public ArrayList<FootballClub> getWinClub() throws IOException {
        ArrayList<FootballClub> clubWin = new ArrayList<>();

        FileInputStream fileInputStream = new FileInputStream("Data.txt");
        ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

        for (; ; ) {
            try {
                FootballClub club = (FootballClub) objectInputStream.readObject();
                clubWin.add(club);
            } catch (EOFException | ClassNotFoundException e) {
                break;
            }
        }
        fileInputStream.close();
        objectInputStream.close();
        int length = clubWin.size();
        for (int i = 0; i < length - 1; i++) {
            for (int j = 0; j < length - (i + 1); j++) {
                if (clubWin.get(j).getWin() < clubWin.get(j + 1).getWin()) {
                    FootballClub part = clubWin.get(j);
                    clubWin.set(j, clubWin.get(j + 1));
                    clubWin.set(j + 1, part);
                }
            }
        }
        return clubWin;

    }

    public ArrayList<FootballClub> getScoreClub() throws IOException {
        ArrayList<FootballClub> clubScore = new ArrayList<>();

        FileInputStream fileInputStream = new FileInputStream("Data.txt");
        ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

        for (; ; ) {
            try {
                FootballClub club = (FootballClub) objectInputStream.readObject();
                clubScore.add(club);
            } catch (EOFException | ClassNotFoundException e) {
                break;
            }
        }
        fileInputStream.close();
        objectInputStream.close();
        int length = clubScore.size();
        for (int i = 0; i < length - 1; i++) {
            for (int j = 0; j < length - (i + 1); j++) {
                if (clubScore.get(j).getScoredGoal() < clubScore.get(j + 1).getScoredGoal()) {
                    FootballClub part = clubScore.get(j);
                    clubScore.set(j, clubScore.get(j + 1));
                    clubScore.set(j + 1, part);
                }
            }
        }
        return clubScore;

    }


}
```

The code of Routes

```
GET     /                       controllers.FrontendController.index()

GET     /clubs                     controllers.LeagueController.getClubs()
GET     /matches                     controllers.LeagueController.getMatches()
GET     /winClubs                    controllers.LeagueController.getClubsWin()
GET     /scoreClub                   controllers.LeagueController.getClubsScore()


GET    /assets/*file              controllers.Assets.versioned(path="/public", file: Asset)
```

Source code of Front end

Code for app component html :-

```
<h1 id="heading">{{heading}}</h1>
<nav id="nav-bar">
   <div id="container">
     <ul>
        <li routerLink="/matchtable"> Match Table </li>
        <li routerLink="/Random">Random Match</li>
        <li> Club Table
           <ul>
              <li routerLink="/clubtable"> Sorted by Point</li>
              <li routerLink="/clubbywin"> Sorted by Win </li>
              <li routerLink="/scoreclub"> Sorted by Score </li>
            </ul>
         </li>

      </ul>
   </div>
</nav>
<router-outlet></router-outlet>
```

Club component ts code :-

```
export class ClubTableComponent implements OnInit {

  Clubs : FootballClub[] = [];


  private httpClient: HttpClient;
  private URL = "http://localhost:9000/clubs";


    constructor (httpClient: HttpClient) {
      this.httpClient = httpClient;

    }
    ngOnInit() {
      this.getList();

    }

    getlistlist():Observable<FootballClub[]>{
      return this.httpClient.get<FootballClub[]>(`${this.URL}`)
  }

  getList(){
   this.getlistlist().subscribe((data:FootballClub[])=>{
    console.log(data);
    this.Clubs=data;
   });

  }


}
```

App components html :-

```
<h2>Points Table</h2>
<div>
<table id="customers">
  <tr>
     <th>Football Club Name</th>
     <th>Football club ID</th>
     <th>Played Matches</th>
     <th>Total Point</th>
     <th>Win</th>
     <th>Defeat</th>
     <th>Draw</th>
     <th>Total Goal</th>

   </tr>
   <tr *ngFor = "let Club of Clubs">
      <td>{{ Club.name}}</td>
      <td>{{ Club.clubID}}</td>
      <td>{{ Club.playedMatches}}</td>
      <td>{{ Club.point}}</td>
      <td>{{ Club.win}}</td>
      <td>{{ Club.defeat}}</td>
      <td>{{ Club.draw}}</td>
      <td>{{ Club.scoredGoal}}</td>
    </tr>

</table>

</div>
```

Match components typescript :-

```
export class MatchTableComponent implements OnInit {


  Matches :Match[] = [];
  searchText;

  private httpClient: HttpClient;
  private URL = "http://localhost:9000/matches";
  day: number;
  month: number;
  year: number;
  Goal1: number;


  constructor (httpClient: HttpClient) {
    this.httpClient = httpClient;

  }
  ngOnInit() {
    this.getList();
  }

  getlistlist():Observable<Match[]>{
    return this.httpClient.get<Match[]>(`${this.URL}`)

  }

  getList(){
    this.getlistlist().subscribe((data:Match[])=>{
      console.log(data);
      this.Matches=data;
    });

  }

}
```

Match components html :-

```
<h2>Match Table</h2>
<div>
    <div id="search-hero">
        <label for="search"> Enter the Date : </label>
        <input id="form-control" type="text" name="search" [(ngModel)]="searchText"
autocomplete="off" placeholder="Enter the date to search">
    </div>
<table id="customers">
    <tr>
      <th>Club No 1</th>
      <th>Club No 2</th>
      <th>Club No 1 Goal</th>
      <th>Club No 2 Goal</th>
      <th>Date</th>
    </tr>
    <tr *ngFor = "let Match of Matches | filter:searchText">
        <td>{{ Match.club1Name }}</td>
        <td>{{ Match.club2Name }}</td>
        <td>{{ Match.club1Goal }}</td>
        <td>{{ Match.club2Goal }}</td>
        <td>{{ Match.stringDate}}</td>
    </tr>
</table>

</div>
```

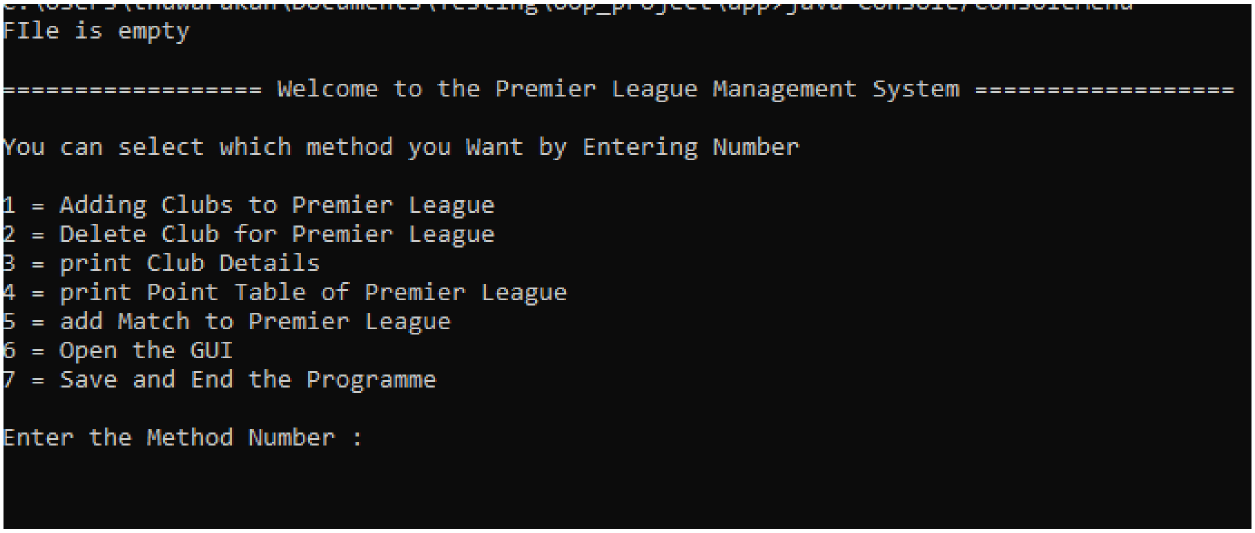
Random component typescript:-

```
export class RandomMatchComponent implements OnInit {

  Clubs : FootballClub[] = [];

  day:number;
  month:number;
  year:number;
  Date:string;
  Goal1:number;
  Goal2:number;
  Club1 : FootballClub;
  Club2 : FootballClub;
  count1 : number;
  count2 : number;


  private httpClient: HttpClient;
private URL = "http://localhost:9000/clubs";



 constructor (httpClient: HttpClient) {
  this.httpClient = httpClient;

 }
 ngOnInit() {
  this.getList();

 }

 getlistlist():Observable<FootballClub[]>{
  return this.httpClient.get<FootballClub[]>(`${this.URL}`)

 }

 getList(){
  this.getlistlist().subscribe((data:FootballClub[])=>{
   console.log(data);
   this.Clubs=data;
  });

 }

 RandomMatch(){
  this.day = Math.floor(Math.random() * (27 + 1)) + 1;
  this.month = Math.floor(Math.random() * (11 + 1)) + 1;
  this.year = 2020;
  this.Goal1 = Math.floor(Math.random() * (9 + 1)) + 1;
  this.Goal2 = Math.floor(Math.random() * (9 + 1)) + 1;
  this.Date = this.day+"/"+this.month+"/"+this.year;
  this.count1 = Math.floor(Math.random() * (this.Clubs.length)) ;
  this.count2 = Math.floor(Math.random() * (this.Clubs.length)) ;
  while(this.count1 == this.count2){
   this.count2 = Math.floor(Math.random() * (this.Clubs.length)) ;
  }
  this.Club2 = this.Clubs[this.count2];
  this.Club1 = this.Clubs[this.count1];


  document.getElementById("Goal1").innerHTML = "First team Goal : "+ this.Goal1;
  document.getElementById("Goal2").innerHTML = "Sceond team Goal : "+ this.Goal2;
  document.getElementById("date").innerHTML = "Date : "+ this.Date;
  document.getElementById("Club1").innerHTML = "First team : "+ this.Club1.name;
  document.getElementById("Club2").innerHTML = "Sceond team : "+ this.Club2.name;
```

```
   let match = new Match();

   let date = new Date();
   date.day = this.day;
   date.month = this.month;
   date.year = this.year;

   match.club1 = this.Club1;
   match.club2 = this.Club2;
   match.club1Goal = this.Goal1;
   match.club2Goal = this.Goal2;
   match.matchDate = date;
   match.stringDate = this.Date;
   match.club1Name = this.Club1.name;
   match.club2Name = this.Club2.name;

  }
}
```

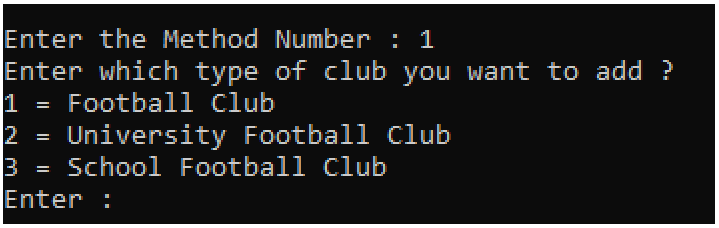
Sample outputs of CLI parts:-

1) Basic look of CLI



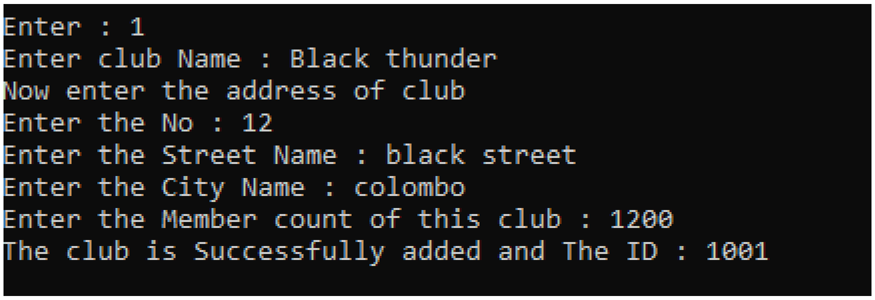
```
C:\Users\thawarakan\Documents\Testing\oop_project\app>java Console/ConsoleMenu
FIle is empty

================== Welcome to the Premier League Management System ==================

You can select which method you Want by Entering Number

1 = Adding Clubs to Premier League
2 = Delete Club for Premier League
3 = print Club Details
4 = print Point Table of Premier League
5 = add Match to Premier League
6 = Open the GUI
7 = Save and End the Programme

Enter the Method Number :
```

02)add club method



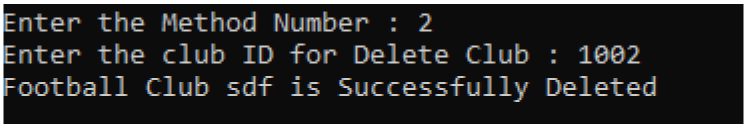
```
Enter the Method Number : 1
Enter which type of club you want to add ?
1 = Football Club
2 = University Football Club
3 = School Football Club
Enter :
```

03) add a football club in CLI



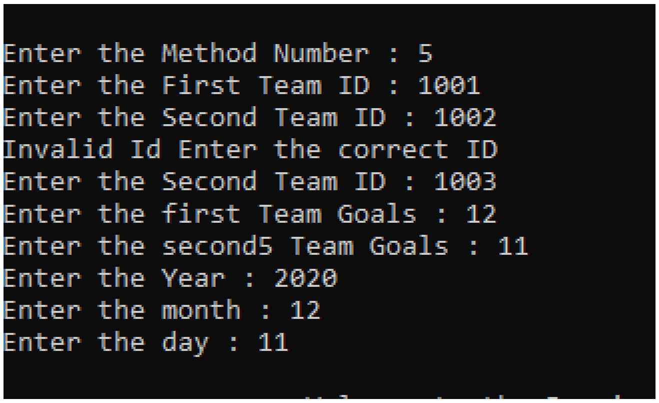
```
Enter : 1
Enter club Name : Black thunder
Now enter the address of club
Enter the No : 12
Enter the Street Name : black street
Enter the City Name : colombo
Enter the Member count of this club : 1200
The club is Successfully added and The ID : 1001
```

04) Delete football club



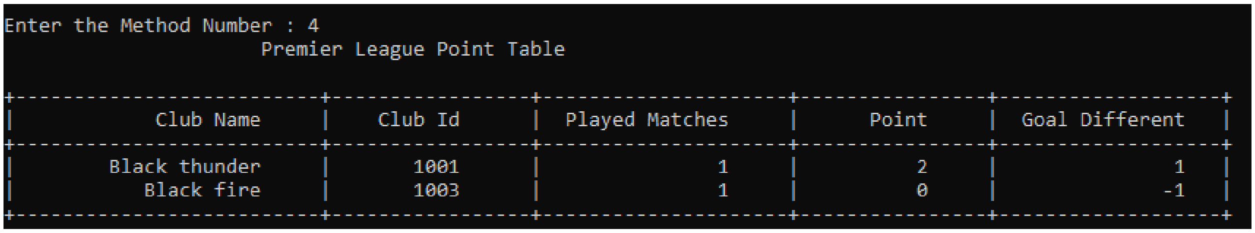
```
Enter the Method Number : 2
Enter the club ID for Delete Club : 1002
Football Club sdf is Successfully Deleted
```

05)add match method



```
Enter the Method Number : 5
Enter the First Team ID : 1001
Enter the Second Team ID : 1002
Invalid Id Enter the correct ID
Enter the Second Team ID : 1003
Enter the first Team Goals : 12
Enter the second5 Team Goals : 11
Enter the Year : 2020
Enter the month : 12
Enter the day : 11
```

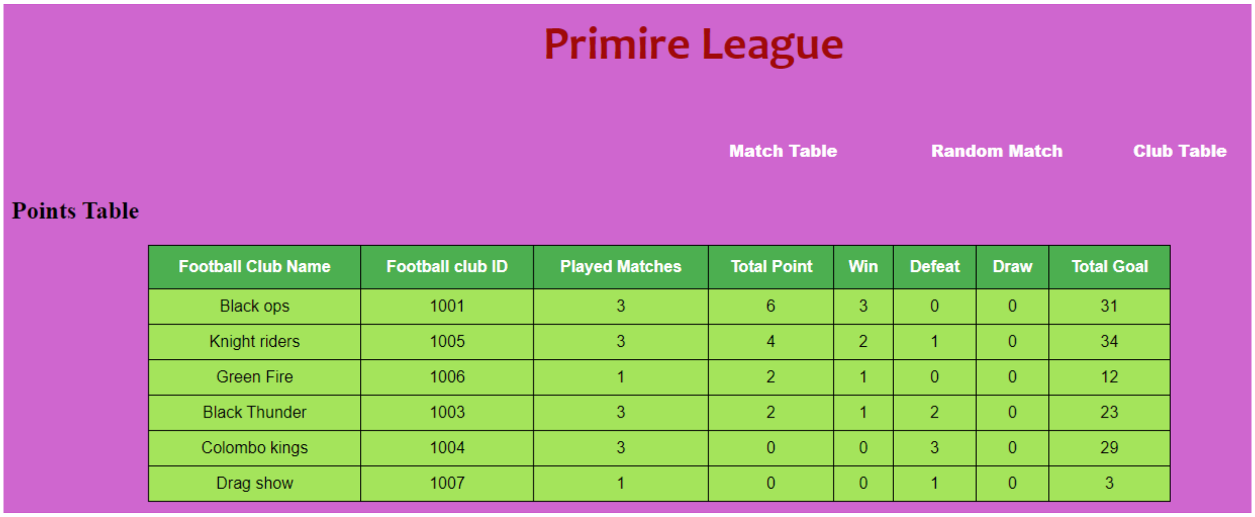
06) point table



```
Enter the Method Number : 4
                      Premier League Point Table

+---------------------------+-----------------+----------------------+-----------------+--------------------+
|             Club Name     |     Club Id     |  Played Matches      |       Point     |  Goal Different    |
+---------------------------+-----------------+----------------------+-----------------+--------------------+
|         Black thunder     |       1001      |                    1 |               2 |                  1 |
|            Black fire     |       1003      |                    1 |               0 |                 -1 |
+---------------------------+-----------------+----------------------+-----------------+--------------------+
```

GUI part –

1) club sort by points

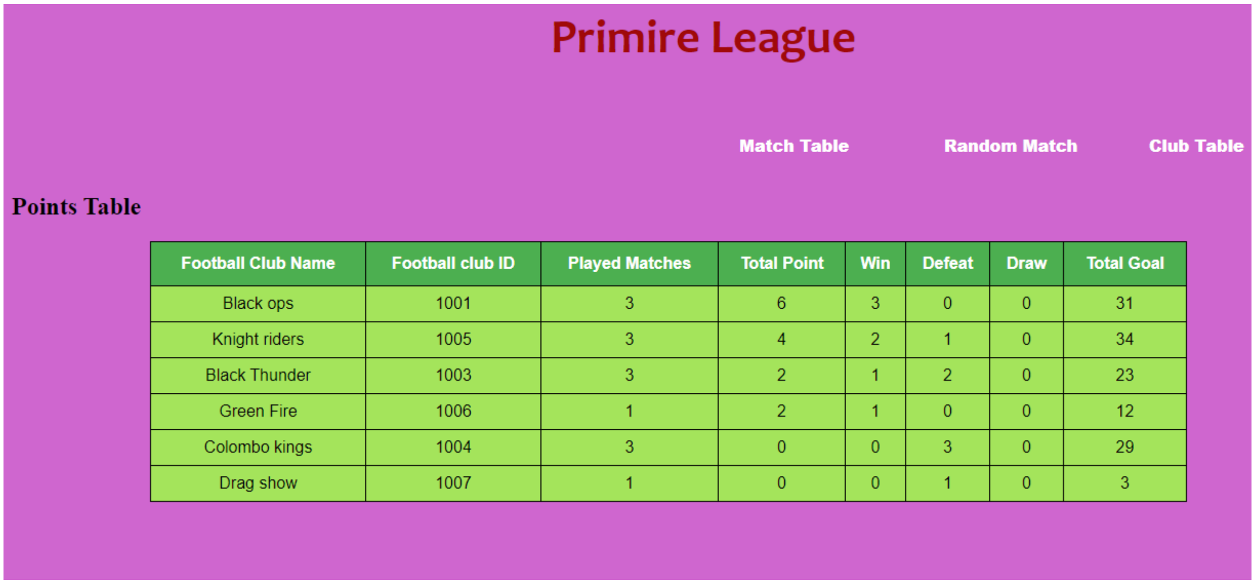


Primire League

Points Table

Football Club Name	Football club ID	Played Matches	Total Point	Win	Defeat	Draw	Total Goal
Black ops	1001	3	6	3	0	0	31
Knight riders	1005	3	4	2	1	0	34
Green Fire	1006	1	2	1	0	0	12
Black Thunder	1003	3	2	1	2	0	23
Colombo kings	1004	3	0	0	3	0	29
Drag show	1007	1	0	0	1	0	3

2) Club sorted by win

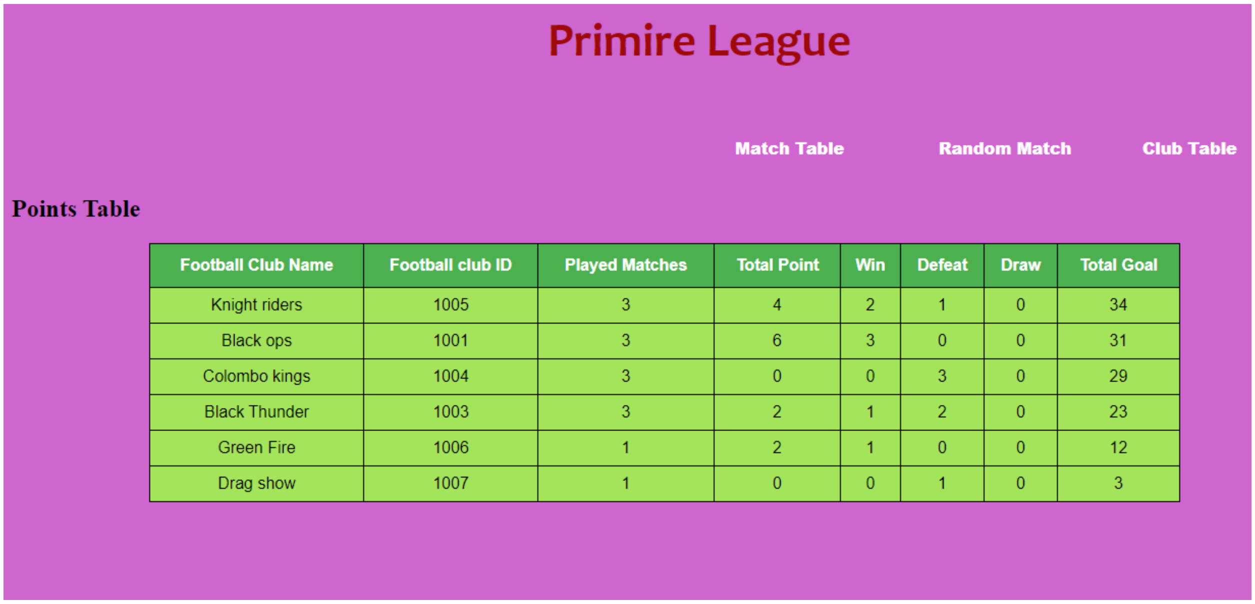


Primire League

Points Table

Football Club Name	Football club ID	Played Matches	Total Point	Win	Defeat	Draw	Total Goal
Black ops	1001	3	6	3	0	0	31
Knight riders	1005	3	4	2	1	0	34
Black Thunder	1003	3	2	1	2	0	23
Green Fire	1006	1	2	1	0	0	12
Colombo kings	1004	3	0	0	3	0	29
Drag show	1007	1	0	0	1	0	3

3) club sort by scored goals

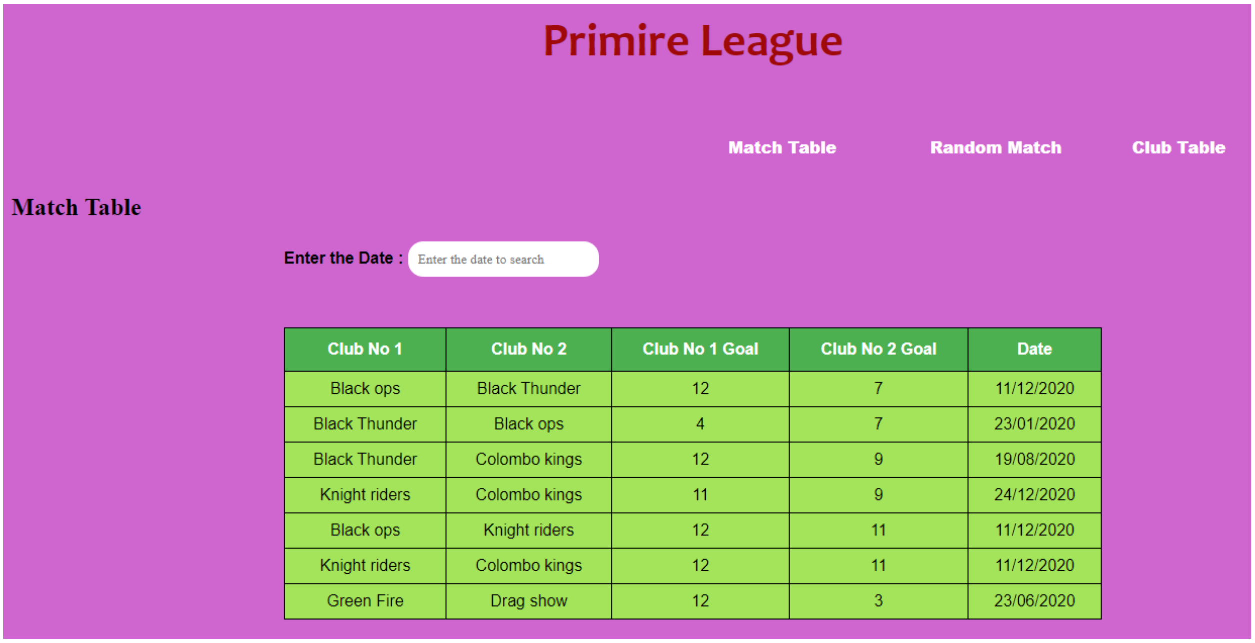


Primire League

Points Table

Football Club Name	Football club ID	Played Matches	Total Point	Win	Defeat	Draw	Total Goal
Knight riders	1005	3	4	2	1	0	34
Black ops	1001	3	6	3	0	0	31
Colombo kings	1004	3	0	0	3	0	29
Black Thunder	1003	3	2	1	2	0	23
Green Fire	1006	1	2	1	0	0	12
Drag show	1007	1	0	0	1	0	3

4) match table



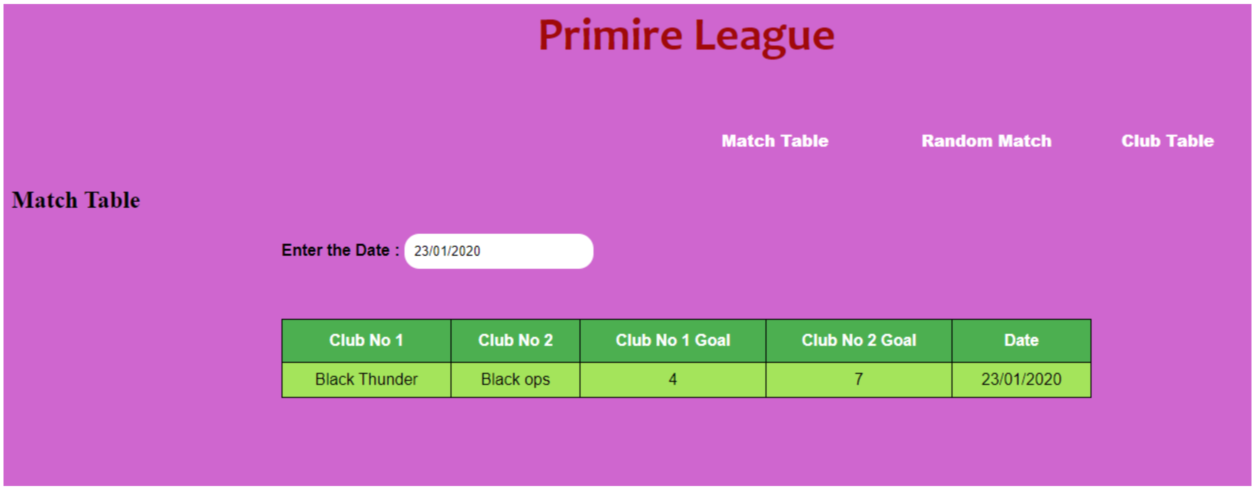
Primire League

Match Table

Enter the Date :

Club No 1	Club No 2	Club No 1 Goal	Club No 2 Goal	Date
Black ops	Black Thunder	12	7	11/12/2020
Black Thunder	Black ops	4	7	23/01/2020
Black Thunder	Colombo kings	12	9	19/08/2020
Knight riders	Colombo kings	11	9	24/12/2020
Black ops	Knight riders	12	11	11/12/2020
Knight riders	Colombo kings	12	11	11/12/2020
Green Fire	Drag show	12	3	23/06/2020

5) match searching method



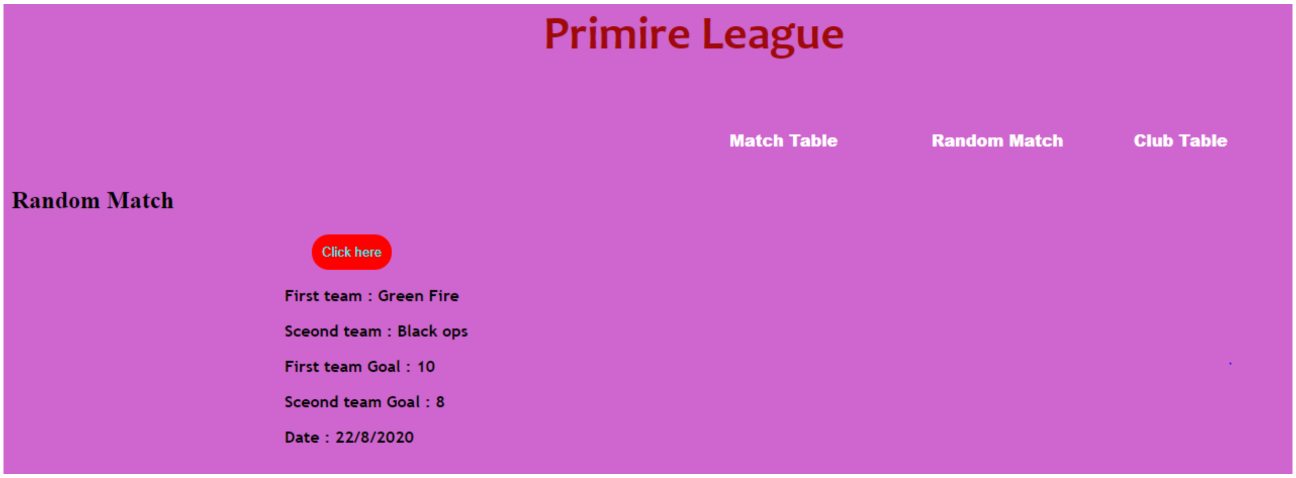
Primire League

Match Table

Enter the Date : 23/01/2020

Club No 1	Club No 2	Club No 1 Goal	Club No 2 Goal	Date
Black Thunder	Black ops	4	7	23/01/2020

6) Randomly create match



Primire League

Random Match

First team : Green Fire

Sceond team : Black ops

First team Goal : 10

Sceond team Goal : 8

Date : 22/8/2020