INFORMATICS INSTITUTE OF TECHNOLOGY

UNIVERSITY OF WESTMINSTER

5COSC007C

Object Oriented Programming

Module Leader: Mr. Guhanathan Poravi

Course Work

Sandeepa Sewwandi Perera
IIT ID - 2017184
W1673745
Tutorial Group A

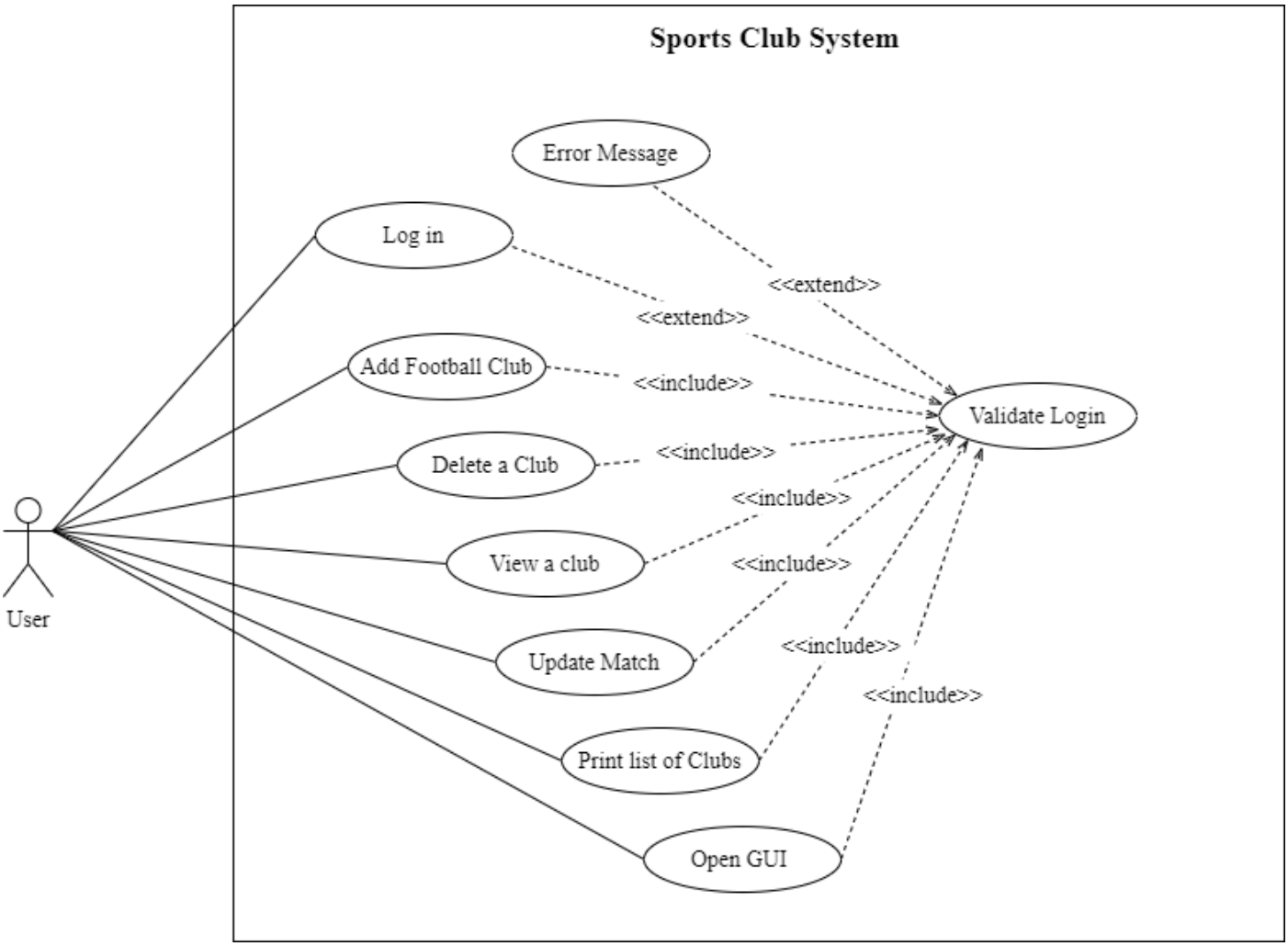
Table of Contents

1. Introduction

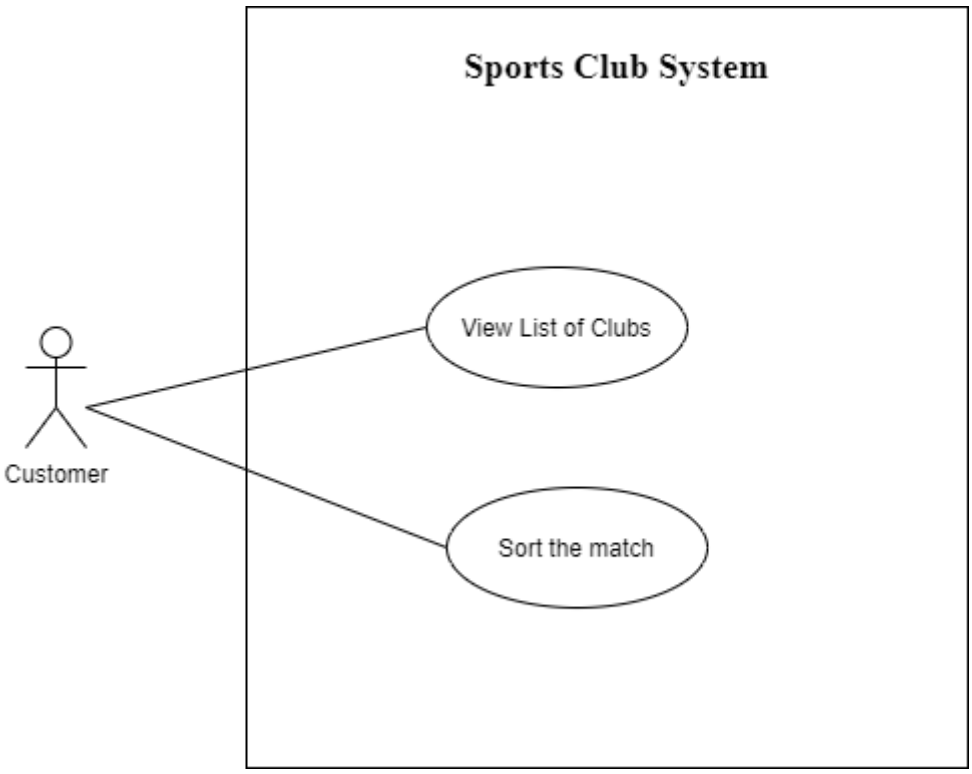
This course work regarding the system of sports unit using football details entering. The system should be used to create sports club, delete, view details about football clubs and the update football matchers. All class codes, interfaces notepad screen shots and command prompt screenshots are attached in this report. The user can view all the teams playing in the premier league and some of their statistics, in descending order, according to the points using this system. Therefore the system should have graphical user interface to view such of details clearly.

2. UML Diagram

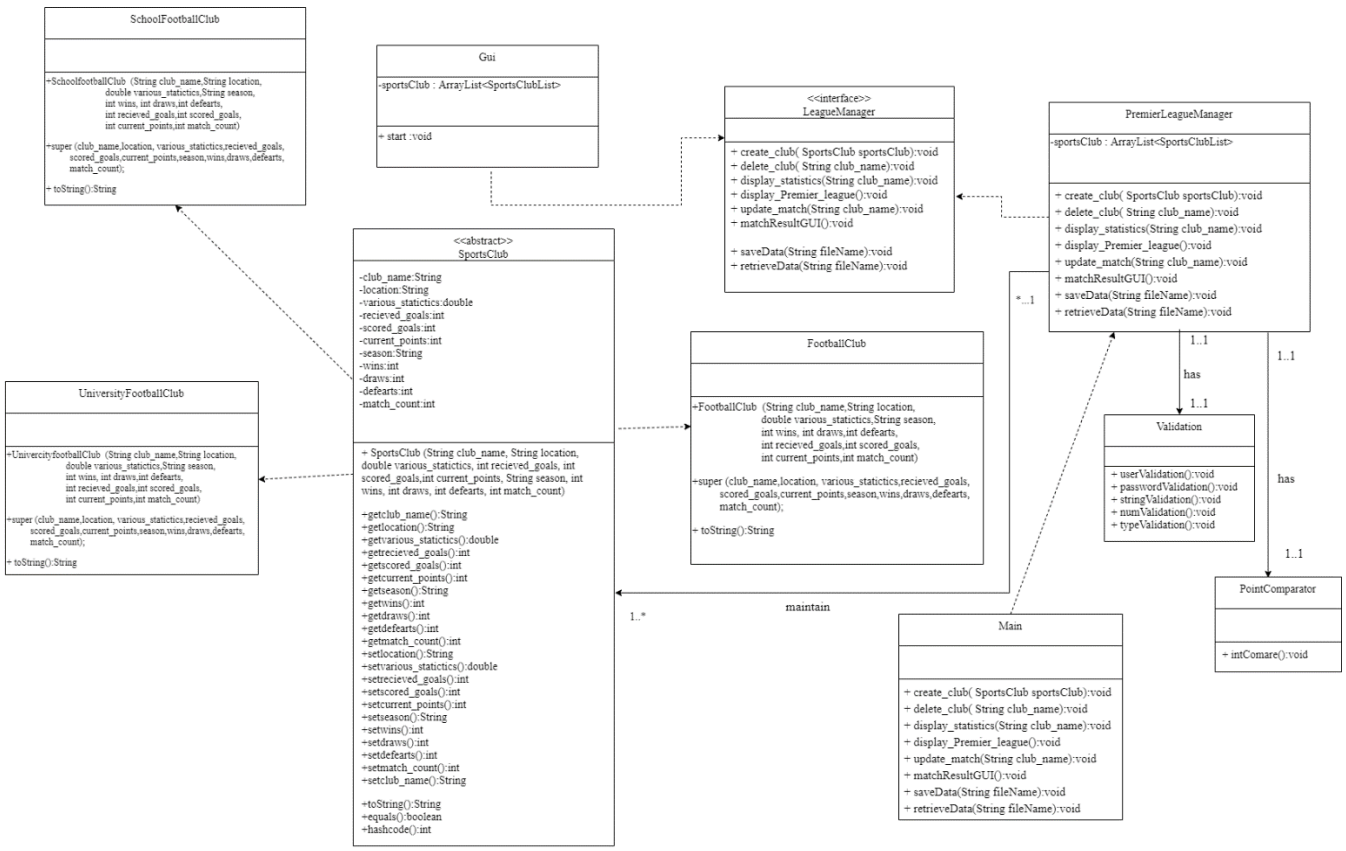
2.1 Use case diagram for Console System



2.2 Use case diagram for GUI



2.3 Class diagram for console system



3. Code related to console system

3.1 SportsClub class (Super class)

```
package lk.iit.oop.pojoclasess;

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

public abstract class SportsClub implements Serializable {

    private String club_name;
    private String location;
    private double various_statictics;
    private int recieved_goals;
    private int scored_goals;
    private int current_points;
    private String season;
    private int wins;
    private int draws;
    private int defearts;
    private int match_count;

    public SportsClub(String club_name, String location, double various_statictics, int
recieved_goals, int scored_goals, int current_points, String season, int wins, int draws,
int defearts, int match_count) {
        this.club_name = club_name;
        this.location = location;
        this.various_statictics = various_statictics;
        this.recieved_goals = recieved_goals;
        this.scored_goals = scored_goals;
        this.current_points = current_points;
        this.season = season;
        this.wins = wins;
        this.draws = draws;
        this.defearts = defearts;
        this.match_count = match_count;
    }

    public SportsClub(){

    }

//getters
    public String getClub_name() {
        return club_name;
    }

    public String getLocation() {
        return location;
    }

    public double getVarious_statictics() {
        return various_statictics;
    }

    public int getRecieved_goals() {
        return recieved_goals;
    }

    public int getScored_goals() {
        return scored_goals;
    }

    public int getCurrent_points() {
        return current_points;
    }

    public String getSeason() {
        return season;
    }
```

```
    public int getWins() {
        return wins;
    }

    public int getDraws() {
        return draws;
    }

    public int getDefearts() {
        return defearts;
    }

    public int getMatch_count() {
        return match_count;
    }

//setters
    public void setClub_name(String club_name) {
        this.club_name = club_name;
    }

    public void setLocation(String location) {
        this.location = location;
    }

    public void setVarious_statictics(double various_statictics) {
        this.various_statictics = various_statictics;
    }

    public void setRecieved_goals(int recieved_goals) {
        this.recieved_goals = recieved_goals;
    }

    public void setScored_goals(int scored_goals) {
        this.scored_goals = scored_goals;
    }

    public void setCurrent_points(int current_points) {
        this.current_points = current_points;
    }

    public void setSeason(String season) {
        this.season = season;
    }

    public void setWins(int wins) {
        this.wins = wins;
    }

    public void setDraws(int draws) {
        this.draws = draws;
    }

    public void setDefearts(int defearts) {
        this.defearts = defearts;
    }

    public void setMatch_count(int match_count) {
        this.match_count = match_count;
    }

    //to string method for print data
    @Override
    public String toString() {
        return club_name +
                location +
                various_statictics +
                recieved_goals +
                scored_goals +
                current_points +
                season +
```

```
                wins +
                draws +
                defearts +
                match_count ;
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        SportsClub that = (SportsClub) o;
        return Double.compare(that.various_statictics, various_statictics) == 0 &&
                recieved_goals == that.recieved_goals &&
                scored_goals == that.scored_goals &&
                current_points == that.current_points &&
                wins == that.wins &&
                draws == that.draws &&
                defearts == that.defearts &&
                match_count == that.match_count &&
                Objects.equals(club_name, that.club_name) &&
                Objects.equals(location, that.location) &&
                Objects.equals(season, that.season);
    }

    @Override
    public int hashCode() {
        return Objects.hash(club_name, location, various_statictics, recieved_goals,
scored_goals, current_points, season, wins, draws, defearts, match_count);
    }


}
```

3.2 FootballClub class

```
package lk.iit.oop.pojoclasess;


public class FootballClub extends SportsClub {

            public FootballClub(String club_name,String location,double
various_statictics, String season, int wins, int draws,int defearts, int
recieved_goals,int scored_goals, int current_points, int match_count){
                super(club_name,location,
various_statictics,recieved_goals,scored_goals,current_points,season,wins,draws,defearts,m
atch_count);
            }

            @Override
            public String toString() {
                return "Normal football Club";
            }

        }
```

3.3 SchoolFootballClub class

```
package lk.iit.oop.pojoclasess;

public class Schoolfootballclub extends SportsClub {
```

```
    public Schoolfootballclub(String club_name,String location,double various_statictics,
String season, int wins, int draws,int defearts, int recieved_goals,int scored_goals, int
current_points, int match_count){
        super(club_name,location,
various_statictics,recieved_goals,scored_goals,current_points,season,wins,draws,defearts,m
atch_count);

    }

    @Override
    public String toString() {
        return "School football Club";
    }

}
```

3.4 UniversityFootballClub class

```
package lk.iit.oop.pojoclasess;

public class Univercityfootballclub extends SportsClub {
   public Univercityfootballclub(String club_name,String location,double various_statictics,
String season, int wins, int draws,int defearts, int recieved_goals,int scored_goals, int
current_points, int match_count){
      super(club_name,location,
various_statictics,recieved_goals,scored_goals,current_points,season,wins,draws,defearts,match_c
ount);
   }

   @Override
   public String toString() {
      return "Univer football Club";
   }

}
```

3.5 LeagueManager (Interface)

```
package lk.iit.oop;
import java.io.IOException;

public interface LeagueManager {
    void create_club( SportsClub sportsClub);
    void delete_club( String club_name);
    void display_statistics(String club_name);
    void display_Premier_league();
    void update_match(String clubOname, String club_name, int pointOne, int pointTwo);
    void matchResultGUI();


    void saveData(String fileName) throws IOException;
    void retrieveData(String fileName) throws IOException, ClassNotFoundException;


}
```

3.5 PremierLeagueManager

```
package lk.iit.oop;
import lk.iit.oop.Exceptions.PointComparator;
import lk.iit.oop.GUI.MatchTableGUI;
import javafx.application.Application;
import lk.iit.oop.pojoclasess.SportsClub;

import java.io.*;
```

```
import java.util.*;

public class PremierLeagueManager implements LeagueManager {

    private ArrayList<SportsClub> sportsClubList = new ArrayList<>();    // array list for sports
club


    @Override
    public void create_club(SportsClub sportsClub) {
        for (SportsClub sportsClub1 : sportsClubList) {
            if (sportsClub.equals(sportsClub1)) {
                System.out.println("club is already in the sport club list");

            }
        }
                sportsClubList.add(sportsClub);
    }


    @Override
    public void delete_club(String club_name) {
        boolean foundclub = false;
        for (SportsClub sportsClub : sportsClubList) {
            if (sportsClub.getClub_name().equals(club_name)) {
                sportsClubList.remove(sportsClub);
                foundclub = true;
                System.out.printf("club was removed successfully!");
                break;
            }
        }
        if (!foundclub) {
            System.out.printf("club is not available\n", club_name);
        }

    }

    @Override
    public void display_statistics(String club_name) {
        if (sportsClubList.isEmpty()) {
            System.out.println("No clubs are in the Sports clubs at the Moment!");
        } else {
            boolean foundclub = false;
            for (SportsClub sportsClub : sportsClubList) {
                if (sportsClub.getClub_name().equals(club_name)) {

                    foundclub = true;
                    System.out.println("  You enterd club -:  " + club_name);
                    System.out.println("-----------------------------------------------------
-------------------------------------------------------------------------------------------
-------------------------------------------");
                    System.out.printf("%10s %25s %15s %25s %20s %20s %20s %10s %10s %15s %15s",
"Club Type", "Club Name", "Location", "Various Statictics", "Recieved Goals", "Scored Goals",
"Current Points", "Wins", "Draws", "defearts", "match_count");
                    System.out.println("   ");
                    System.out.println("-----------------------------------------------------------
-------------------------------------------------------------------------------------------
-------------------------------------------");

                    Collections.sort(sportsClubList, new PointComparator().reversed());

                    System.out.format("%15s %12s %15s %20s %20s %22s %20s %15s %9s %13s %12" +
                                    "s",
                            sportsClub, sportsClub.getClub_name(), sportsClub.getLocation(),
sportsClub.getVarious_statictics(), sportsClub.getRecieved_goals(),
sportsClub.getScored_goals(), sportsClub.getCurrent_points(), sportsClub.getWins(),
sportsClub.getDraws(), sportsClub.getDefearts(), sportsClub.getMatch_count());
                    System.out.println();
                    break;
                }
            }
            if (!foundclub) {
```

```
                System.out.printf("club is not available\n", club_name);
            }

        }
    }



    @Override    //print list of clubs
    public void display_Premier_league() {
       if (sportsClubList.isEmpty()) {
            System.out.println("no clubs found");
            return;
        }
        System.out.println("                                                                    List of all
clubs currently available");
        System.out.println("    ");
        System.out.println("-------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
--------------------------------");
        System.out.printf("%10s %25s %15s %25s %20s %20s %20s %10s %10s %15s %15s", "Club Type",
"Club Name", "Location", "Various Statictics", "Recieved Goals", "Scored Goals", "Current
Points","Wins","Draws","defearts","match_count");
        System.out.println("    ");
        System.out.println("-------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
--------------------------------");
         Collections.sort(sportsClubList, new PointComparator().reversed());

        for (SportsClub sportsClub : sportsClubList) {
            System.out.format("%15s %12s %15s %20s %20s %22s %20s %15s %9s %13s %12" +
                              "s",

sportsClub,sportsClub.getClub_name(),sportsClub.getLocation(),sportsClub.getVarious_statictics()
,sportsClub.getRecieved_goals(),sportsClub.getScored_goals(),sportsClub.getCurrent_points(),spor
tsClub.getWins(),sportsClub.getDraws(),sportsClub.getDefearts(),sportsClub.getMatch_count());
            System.out.println();

        }
        System.out.println("-------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------
--------------------------------");
    }


    @Override
    public void update_match(String club_name, String clubTname, int pointOne, int pointTwo) {
        boolean foundclub = false;
        for (SportsClub sportsClub : sportsClubList) {
            if (sportsClub.getClub_name().equals(club_name)) {

                //sportsClubList.update(sportsClub);

                foundclub = true;
                System.out.printf("\t club was updated successfully!");
                break;
            }
        }
        if (!foundclub) {
            System.out.printf("club is not available\n", club_name);
        }
    }

    @Override
    public void matchResultGUI() {
        Application.launch(MatchTableGUI.class);

    }



    public void saveData(String fileName) throws IOException {
```

```
        FileOutputStream fileOutputStream = new FileOutputStream("saveFile.txt");
        ObjectOutputStream objectOutputStream = new ObjectOutputStream(fileOutputStream);

        objectOutputStream.writeObject(sportsClubList);

        System.out.println("club have been saved successfully");
    }

    public void retrieveData(String fileName) throws IOException {
        FileInputStream fileInputStream = new FileInputStream("saveFile.txt");
        ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

        try {
            sportsClubList = (ArrayList<SportsClub>) objectInputStream.readObject();

        } catch (Exception e) {

        }
        System.out.println("Vehicles have been loaded successfully");



    }
}
```

3.6 Main class

```
package lk.iit.oop;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;
import lk.iit.oop.Exceptions.Validation;
import lk.iit.oop.pojoclasess.FootballClub;
import lk.iit.oop.pojoclasess.Schoolfootballclub;
import lk.iit.oop.pojoclasess.SportsClub;
import lk.iit.oop.pojoclasess.Univercityfootballclub;

import java.io.IOException;
import java.util.Scanner;

import static javafx.application.Application.launch;


public class Main {
    private static LeagueManager manager = new PremierLeagueManager();
    static Scanner in = new Scanner(System.in).useDelimiter("\n");

    public static void create_club() {

        SportsClub sportsClub = null;

        System.out.println("Enter the SportsClub name");
        String clubname= Validation.stringValidation(in);

        System.out.println("Enter Location");
        String location = Validation.stringValidation(in);

        System.out.println("Enter the various statictics");
         int staticstic = Integer.parseInt(Validation.numValidation(in));


        while (true) {
            System.out.println("\nSelect your club type: Normal, School or University \n " +
                    "    a).  Normal Football Club \n" +                    // asking from the
user to add a member type
                    "     b).  School Football Club \n" +
                    "     c).  University Football Club");
```

```
            String clubType = Validation.typeValidation(in);

            System.out.println("Enter the Season");
            String season = Validation.stringValidation(in);

            System.out.println("Enter the Wins");
            int wins = Integer.parseInt(Validation.numValidation(in));

            System.out.println("Enter the Draws");
            int draws = Integer.parseInt(Validation.numValidation(in));

            System.out.println("Enter the Defearts");
            int defearts = Integer.parseInt(Validation.numValidation(in));

            System.out.println("Enter the Recieved Goals");
            int rgoals = Integer.parseInt(Validation.numValidation(in));

            System.out.println("Enter the Scored Goals");
            int sgoals = Integer.parseInt(Validation.numValidation(in));

            System.out.println("Enter the Current Points");
            int points = Integer.parseInt(Validation.numValidation(in));

            System.out.println("Enter the Match Count");
            int counts = Integer.parseInt(Validation.numValidation(in));



            switch (clubType) {  //transfer data to club type
                case "a":
                    sportsClub = new FootballClub(clubname, location, staticstic, season, wins,
draws, defearts, rgoals, sgoals, points, counts);
                    break;
                case "b":
                    sportsClub = new Schoolfootballclub(clubname, location, staticstic, season,
wins, draws, defearts, rgoals, sgoals, points, counts);
                    break;
                case "c":
                    sportsClub = new Univercityfootballclub(clubname, location, staticstic,
season, wins, draws, defearts, rgoals, sgoals, points, counts);
                    break;
                default:
                    System.out.println("Invalid Option!!! Please re-enter...");
                    continue;
            }

            manager.create_club(sportsClub);   // added data
            System.out.println("Sport club added sucessfully");

            break;
        }
    }


    public static void main(String[] args) throws IOException, ClassNotFoundException {
        manager.retrieveData("saveFile.txt"); //lodad data from save file

        menuloop:  // welcome note and the user validation
        // I named it menuloop

        System.out.println("                                                          \n");
        System.out.println(" ********* Welcome to the Sports Club  ********* \n");
        System.out.println("                                                      ");
        System.out.println("      ");
        System.out.print("Enter the User name  -:  ");
        String username = Validation.usernameValidation(in);
        System.out.print("Enter the Password    -: ");
        int counts = Integer.parseInt(Validation.passwordValidation(in));

        menu:
        while (true) {
            // main menu of the system
```

```
            System.out.println("      ");
            System.out.println("\t\tWelcome to the Menu");
            System.out.println("      ");
            System.out.println("\tPress 1 to add a SportsClub");
            System.out.println("\tPress 2 to delete the club details");
            System.out.println("\tPress 3 to Display selected club");
            System.out.println("\tPress 4 to print the list of clubs");
            System.out.println("\tPress 5 to Update match");
            System.out.println("\tPress 6 to GUI");
            System.out.println("\tPress 7 to exit");

            Scanner sc = new Scanner(System.in);   //getting user choices
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    create_club();   //add club to the list
                    break;

                case 2:
                    delete_club();   //deleate selected club
                    break;

                case 3:
                    displayvariousstatictics(); // display selected club details
                    break;

                case 4:
                    displayPremierLeague();   //display all the clubs
                    break;

                case 5:
                    update_match();    //update club results
                    break;

                case 6:
                    matchResultGUI();    //open the GUI
                    break ;

                case 7:
                    manager.saveData("saveFile.txt");    // save data to txt file
                    break menu;
                default:
                    System.out.println("Choice is invalid!!! Please re-enter...");
            }
        }

    }
    private static void matchResultGUI() {
        manager.matchResultGUI();
    }


    private static void update_match() {  //updating match results
        Scanner sc = new Scanner(System.in);
        System.out.println("Please enter the name of the club One you want to update:");
        String clubOname = sc.nextLine();

        System.out.println("Please enter the points:");
        int pointOne = sc.nextInt();

        System.out.println("Please enter the name of the club Two you want to update:");
        String clubTname = sc.next();

        System.out.println("Please enter the points:");
        int pointTwo = sc.nextInt();

        manager.update_match(clubOname,clubTname,pointOne,pointTwo);  //sent data to update_club
in PremierleagueManager class
    }
```

```
    private static void delete_club() {
        System.out.println("Please enter the name of the club you want to remove:");
        Scanner sc = new Scanner(System.in);
        String clubname = sc.nextLine();
        manager.delete_club(clubname); //sent data to delete_club method in PremierLeagueManager
class
    }

    private static void displayvariousstatictics() {
        System.out.println("club name to view");
        Scanner sc = new Scanner(System.in);
        String clubname = sc.nextLine();
        manager.display_statistics(clubname); //display club from list
    }

    private static void displayPremierLeague() {

        manager.display_Premier_league();
    }

}
```

3.6 MatchTableGUI

```
package lk.iit.oop.GUI;

import lk.iit.oop.PremierLeagueManager;
import lk.iit.oop.pojoclasess.SportsClub;
import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.scene.text.Font;
import javafx.stage.Stage;

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


public class MatchTableGUI extends Application {
    private static PremierLeagueManager LeagueManager;
    private TableView<SportsClub> tableMember;
    private TextField filterMember;
    private ObservableList<SportsClub> sportsClubList = FXCollections.observableArrayList();


    @Override
    public void start(Stage primaryStage) throws Exception {
        LeagueManager = new PremierLeagueManager();
        System.out.println("...Match Results...");
        primaryStage.setTitle("Sports Club System - Football");
        primaryStage.setHeight(500);
        primaryStage.setWidth(1225);


        TableColumn<SportsClub, String> clubnameColumn = new TableColumn<>("Club Name");
        clubnameColumn.setMinWidth(121);
        clubnameColumn.setCellValueFactory(new PropertyValueFactory<>("club_name"));

        TableColumn<SportsClub, String> locationColumn = new TableColumn<>("Location");
        locationColumn.setMinWidth(125);
```

```
        locationColumn.setCellValueFactory(new PropertyValueFactory<>("Location"));

        TableColumn<SportsClub, String> various_staticticsColumn = new TableColumn<>("Various
Statictics");
        various_staticticsColumn.setMinWidth(130);
        various_staticticsColumn.setCellValueFactory(new
PropertyValueFactory<>("various_statictics"));

        TableColumn<SportsClub, String> recievedgoalsColumn = new TableColumn<>("Recieved
Goals");
        recievedgoalsColumn.setMinWidth(110);
        recievedgoalsColumn.setCellValueFactory(new PropertyValueFactory<>("recieved_goals"));

        TableColumn<SportsClub, String> scoredgoalsColumn = new TableColumn<>(" Scored Goals");
        scoredgoalsColumn.setMinWidth(110);
        scoredgoalsColumn.setCellValueFactory(new PropertyValueFactory<>("scored_goals"));

        TableColumn<SportsClub, String> currentpointsColumn = new TableColumn<>("Current
Points");
        currentpointsColumn.setMinWidth(121);
        currentpointsColumn.setCellValueFactory(new PropertyValueFactory<>("current_points"));

        TableColumn<SportsClub, String> winsColumn = new TableColumn<>("Wins");
        winsColumn.setMinWidth(119);
        winsColumn.setCellValueFactory(new PropertyValueFactory<>("wins"));

        TableColumn<SportsClub, String> drawsColumn = new TableColumn<>("Draws");
        drawsColumn.setMinWidth(119);
        drawsColumn.setCellValueFactory(new PropertyValueFactory<>("Draws"));

        TableColumn<SportsClub, String> defeartsColumn = new TableColumn<>("Defearts");
        defeartsColumn.setMinWidth(119);
        defeartsColumn.setCellValueFactory(new PropertyValueFactory<>("Defearts"));

        TableColumn<SportsClub, String> matchcountColumn = new TableColumn<>("Match Count");
        matchcountColumn.setMinWidth(119);
        matchcountColumn.setCellValueFactory(new PropertyValueFactory<>("match_count"));


        filterMember = new TextField();
        filterMember.setPromptText("Enter name or ID");
        filterMember.setMinWidth(100);

        Label searchLabel = new Label("Search Member: ");
        searchLabel.setFont(new Font(17));

        HBox hBox = new HBox();
        hBox.setPadding(new Insets(15, 11, 11, 60));
        hBox.setSpacing(11);
        hBox.getChildren().addAll(searchLabel, filterMember);

        tableMember = new TableView<>();
        tableMember.setItems(getSportsClubList());
        tableMember.getColumns().addAll(clubnameColumn, locationColumn,
various_staticticsColumn, recievedgoalsColumn, scoredgoalsColumn, currentpointsColumn,
winsColumn, drawsColumn, defeartsColumn, matchcountColumn);


        VBox vBox = new VBox(5);
        Label label = new Label("Sports Club - Football Club System");
        label.setFont(new Font(20));
        vBox.getChildren().addAll(label, tableMember, hBox);
        Scene scene = new Scene(vBox);
        tableMember.setEditable(true);

        vBox.setPadding(new Insets(17, 11, 15, 11));

       // FilteredList<SportsClub> filteredMemberList = new FilteredList<>(sportsClubList, e ->
true);
         filterMember.setOnKeyPressed(e -> {

      //  filterMember.textProperty().addListener(((observable, oldValue, newValue) -> {
```

```
          /*  filteredMemberList.setPredicate((Predicate<? super SportsClub>) sportsClubList ->
{
                if (e.getText() == null || e.getText().isEmpty()) {
                    return true;
                }
                String valueLowerCase = e.getText().toLowerCase();
                if (sportsClubList.getClub_name().toLowerCase().contains(valueLowerCase)) {
                    return true;
                }
                return false;
            });*/
   //     }
     //   ));
          ObservableList<SportsClub> searchResults = FXCollections.observableArrayList();
          for(SportsClub sportsClub:sportsClubArrayList){
              if(sportsClub.getClub_name().toLowerCase().contains(e.getText().toLowerCase())){
                  searchResults.add(sportsClub);
              }
          }
        // SortedList<SportsClub> searchResults = new SortedList<>(filteredMemberList);
        // searchResults.comparatorProperty().bind(tableMember.comparatorProperty());
          tableMember.setItems( searchResults);

          });

          primaryStage.setScene(scene);
          primaryStage.show();
      }

      ArrayList<SportsClub> sportsClubArrayList = new ArrayList<>();

      public ObservableList<SportsClub> getSportsClubList() throws IOException {
          FileInputStream fileInputStream = new FileInputStream("saveFile.txt");
          ObjectInputStream objectInputStream = new ObjectInputStream(fileInputStream);

          try {
              sportsClubArrayList = (ArrayList<SportsClub>) objectInputStream.readObject();

          } catch (Exception e) {

          }
          System.out.println("Clubs have been loaded successfully");
          ObservableList<SportsClub> sportsclubs =
FXCollections.observableArrayList(sportsClubArrayList);
          return sportsclubs;
      }
}
```

3.7 Validation

```
package lk.iit.oop.Exceptions;

import java.util.InputMismatchException;
import java.util.Scanner;

public class Validation {
    public static String usernameValidation(Scanner in) {     //User input validation
        while (true) {                                        //while loop to run continuously if
the input is wrong
            try {
                String userIn = in.next();
                if (!userIn.matches("user")) {        //regular expression to validate through
numbers
                    System.out.println("Wrong User Name! Re enter correct one");

                } else if (userIn.length() == 0) {     //checking whether user input is empty
                    System.out.println("You need to enter a value (not Blank), Enter Again: ");

                } else {
                    return userIn;
```

```
                }

            } catch (Exception e) {     //handliing exception
                numValidation(in);
            }
        }
    }

    public static String passwordValidation(Scanner in) {     //User input validation
        while (true) {                                        //while loop to run continuously if
the input is wrong
            try {
                String userIn = in.next();
                if (!userIn.matches("123")) {        //regular expression to validate through
numbers
                    System.out.println("Wrong Password! Re enter correct one");

                } else if (userIn.length() == 0) {     //checking whether user input is empty
                    System.out.println("You need to enter a value (not Blank), Enter Again: ");

                } else {
                    return userIn;
                }

            } catch (Exception e) {     //handliing exception
                numValidation(in);
            }
        }
    }

    public static String stringValidation(Scanner in) {     //String input validation
        while (true) {                                      //while loop to run continuously if
the input is wrong
            try {
                String userIn = in.next();
                if (!userIn.matches("[a-zA-Z ]+")) {        //regular expression to validate
through numbers
                    System.out.println("Not a valid value..Enter again using only letters!");

                } else if (userIn.length() == 0) {     //checking whether user input is empty
                    System.out.println("You need to enter a value (not Blank), Enter Again: ");

                } else {
                    return userIn;
                }

            } catch (Exception e) {     //handliing exception
                numValidation(in);
            }
        }
    }


    public static String numValidation(Scanner in) {         //validation of integer numbers
        while (true) {
            try {
                String userIn = in.next();
                if (userIn.matches("[^0-9]+")) {    //validation to only allow numbers using
REGEX
                    System.out.println("Not a valid value..Enter again using only Numbers!");

                } else if (userIn.length() == 0) {         //checking the length of the input
                    System.out.println("You need to enter a value (not Blank), Enter Again: ");

                } else {
                    return userIn;
                }

            } catch (Exception e) {
                numValidation(in);
            }
        }
```

```
    }


    public static String typeValidation(Scanner in) {     //String input validation
        while (true) {                                      //while loop to run continuously if
the input is wrong
            try {
                String userIn = in.next();
                if (!userIn.matches("[a-cA-C ]+")) {        //regular expression to validate
through numbers
                    System.out.println("Not a valid value..Enter again using only letters!");

                } else if (userIn.length() == 0) {     //checking whether user input is empty
                    System.out.println("You need to enter a value (not Blank), Enter Again: ");

                } else {
                    return userIn;
                }

            } catch (Exception e) {     //handliing exception
                numValidation(in);
            }
        }
    }

}
```

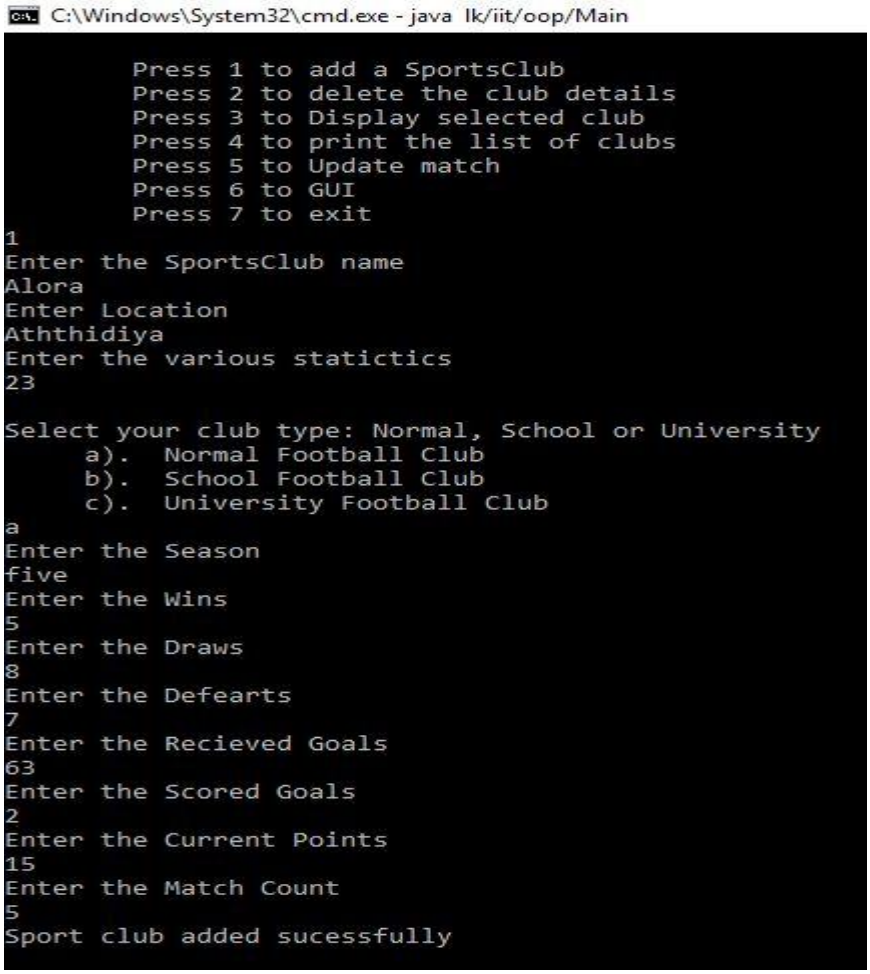
3.8 PointComparator

```
package lk.iit.oop.Exceptions;

import lk.iit.oop.pojoclasess.SportsClub;

import java.util.Comparator;

public class PointComparator  implements Comparator<SportsClub> {

    @Override
    public int compare(SportsClub o1, SportsClub o2) {

        int point = o1.getCurrent_points() - (o2.getCurrent_points());
        if (point != point) {
            return o1.getCurrent_points() - (o2.getCurrent_points());
        }
        else {
            int rg = o1.getRecieved_goals() - (o2.getRecieved_goals());
            if (rg != rg) {
                return o1.getRecieved_goals() - o2.getRecieved_goals();
            }
            else {
                int sg = (o1.getScored_goals()-o2.getScored_goals());
                if (sg != sg){
                    return o2.getScored_goals()-o1.getScored_goals();
                }
            }




        return o1.getCurrent_points() - (o2.getCurrent_points());

    }
    }
}
```

4. Screen shots of CMD

4.1 Menu

4.2 Add Sports Club



4.3 Delete selected club

C:\Windows\System32\cmd.exe - java lk/iit/oop/Main

```
 Club Type                Club Name        Location    V_Stati     Rec_Goals      Scor_Goals   C_Points       Wins      Draws   defearts     match_count
-----------------------------------------------------------------------------------------------------------------------------------------------------------
Univer football Club Pera Champions           Kandy        5.0         4            6          89         10        12           15          10
Normal football Club              b     wallawaththa        6.0         2           36          56          5         6            3           3
Normal football Club         Smilers         Gampaha        8.0         9            8          45          8         6            5           6
School football Club              a         colombo       25.0         8            9          36          5         6            3           5
Normal football Club          Happy         colombo        5.0        96           23          36         56        36           56          10
School football Club            PCC       Pinnawala        8.0        45           63          12         45        63           25           5
-----------------------------------------------------------------------------------------------------------------------------------------------------------

               Welcome to the Menu

       Press 1 to add a SportsClub
       Press 2 to delete the club details
       Press 3 to Display selected club
       Press 4 to print the list of clubs
       Press 5 to Update match
       Press 6 to GUI
       Press 7 to exit
2
Please enter the name of the club you want to remove:
a
club was removed successfully!
               Welcome to the Menu

       Press 1 to add a SportsClub
       Press 2 to delete the club details
       Press 3 to Display selected club
       Press 4 to print the list of clubs
       Press 5 to Update match
       Press 6 to GUI
       Press 7 to exit
4
                                              List of all clubs currently available

-----------------------------------------------------------------------------------------------------------------------------------------------------------
 Club Type                Club Name        Location    V_Stati     Rec_Goals      Scor_Goals   C_Points       Wins      Draws   defearts     match_count
-----------------------------------------------------------------------------------------------------------------------------------------------------------
Univer football Club Pera Champions           Kandy        5.0         4            6          89         10        12           15          10
Normal football Club              b     wallawaththa        6.0         2           36          56          5         6            3           3
Normal football Club         Smilers         Gampaha        8.0         9            8          45          8         6            5           6
Normal football Club          Happy         colombo        5.0        96           23          36         56        36           56          10
School football Club            PCC       Pinnawala        8.0        45           63          12         45        63           25           5
-----------------------------------------------------------------------------------------------------------------------------------------------------------
```

4.4 View selected club

C:\Windows\System32\cmd.exe - java lk/iit/oop/Main

```
-----------------------------------------------------------------------------------------------------------------------------------------------------------

               Welcome to the Menu

       Press 1 to add a SportsClub
       Press 2 to delete the club details
       Press 3 to Display selected club
       Press 4 to print the list of clubs
       Press 5 to Update match
       Press 6 to GUI
       Press 7 to exit
3
club name to view
b
  You enterd club -:  b
-----------------------------------------------------------------------------------------------------------------------------------------------------------
 Club Type                Club Name        Location    V_Stati     Rec_Goals      Scor_Goals   C_Points       Wins      Draws   defearts     match_count
-----------------------------------------------------------------------------------------------------------------------------------------------------------
Normal football Club              b     wallawaththa        6.0         2           36          56          5         6            3           3
```

4.5 View list of clubs

C:\Windows\System32\cmd.exe - java lk/iit/oop/Main

```
C:\Users\SANDEEPA PERERA\Documents\SportsClubSystem\src>java lk/iit/oop/Main
Vehicles have been loaded successfully


 ********* Welcome to the Sports Club  *********



Enter the User name  -:  user
Enter the Password    -: 123

               Welcome to the Menu

       Press 1 to add a SportsClub
       Press 2 to delete the club details
       Press 3 to Display selected club
       Press 4 to print the list of clubs
       Press 5 to Update match
       Press 6 to GUI
       Press 7 to exit
4
                                              List of all clubs currently available

-----------------------------------------------------------------------------------------------------------------------------------------------------------
 Club Type                Club Name        Location    V_Stati     Rec_Goals      Scor_Goals   C_Points       Wins      Draws   defearts     match_count
-----------------------------------------------------------------------------------------------------------------------------------------------------------
Univer football Club Pera Champions           Kandy        5.0         4            6          89         10        12           15          10
Normal football Club              b     wallawaththa        6.0         2           36          56          5         6            3           3
Normal football Club         Smilers         Gampaha        8.0         9            8          45          8         6            5           6
School football Club              a         colombo       25.0         8            9          36          5         6            3           5
Normal football Club          Happy         colombo        5.0        96           23          36         56        36           56          10
School football Club            PCC       Pinnawala        8.0        45           63          12         45        63           25           5
-----------------------------------------------------------------------------------------------------------------------------------------------------------
```

4.6 Update match results

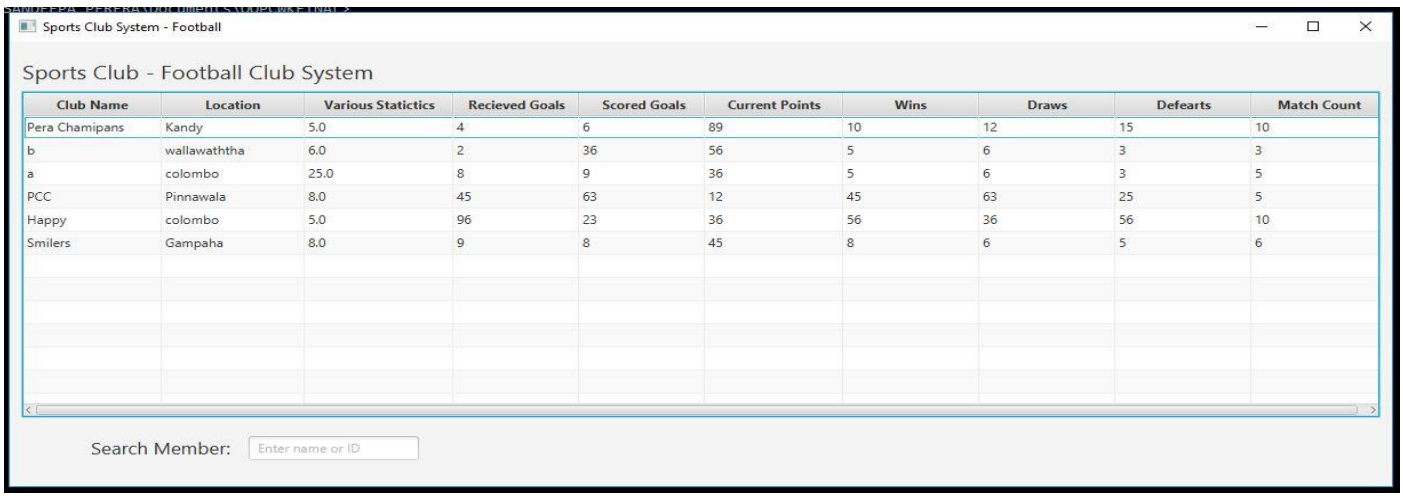
```
C:\Windows\System32\cmd.exe - java  lk/iit/oop/Main
                  Welcome to the Menu

         Press 1 to add a SportsClub
         Press 2 to delete the club details
         Press 3 to Display selected club
         Press 4 to print the list of clubs
         Press 5 to Update match
         Press 6 to GUI
         Press 7 to exit
5
Please enter the name of the club One you want to update:
PCC
Please enter the points:
65
Please enter the name of the club Two you want to update:
b
Please enter the points:
20
          club was updated successfully!
```

4.7 validation

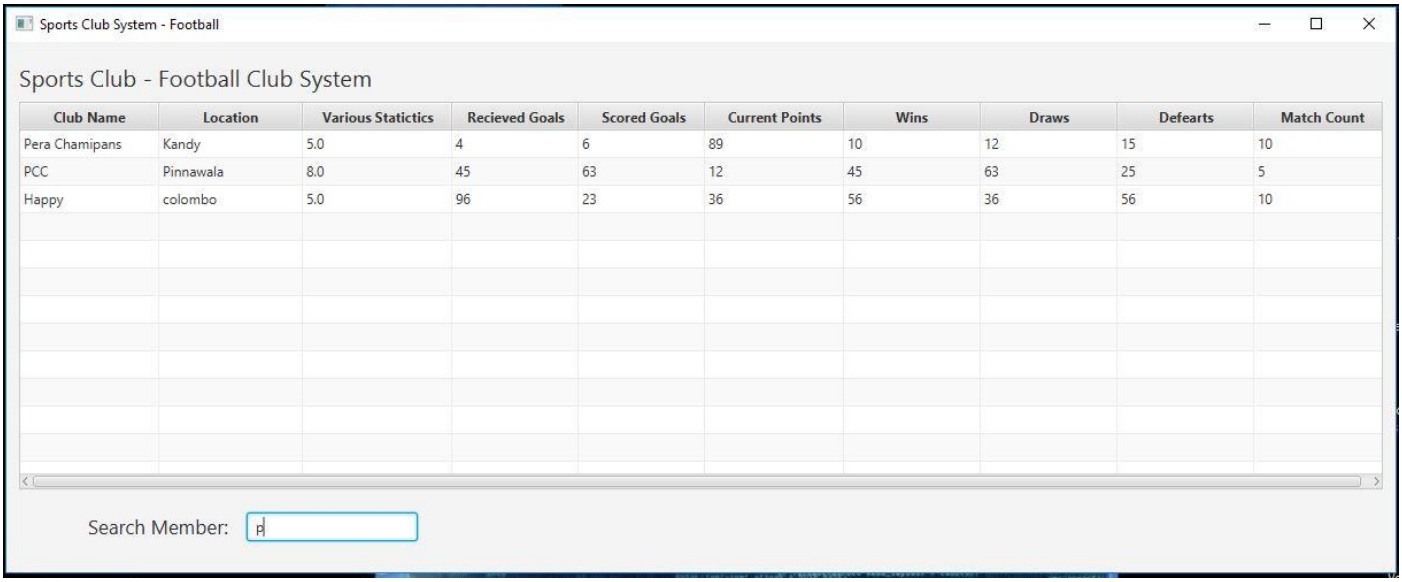
```
C:\Windows\System32\cmd.exe - java  lk/iit/oop/Main
 ********* Welcome to the Sports Club  *********


Enter the User name  -:  user
Enter the Password    -: asa
Wrong Password! Re enter correct one
1562
Wrong Password! Re enter correct one
123

               Welcome to the Menu

       Press 1 to add a SportsClub
       Press 2 to delete the club details
       Press 3 to Display selected club
       Press 4 to print the list of clubs
       Press 5 to Update match
       Press 6 to GUI
       Press 7 to exit
```

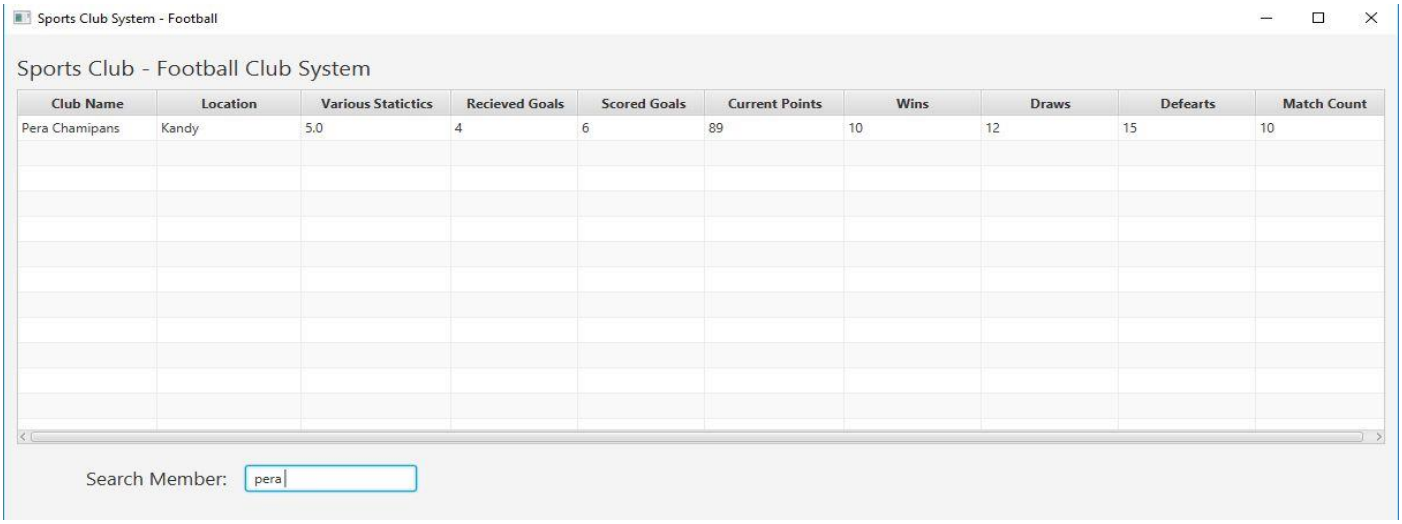
5. Screen shots of GUI

5.1 View Table



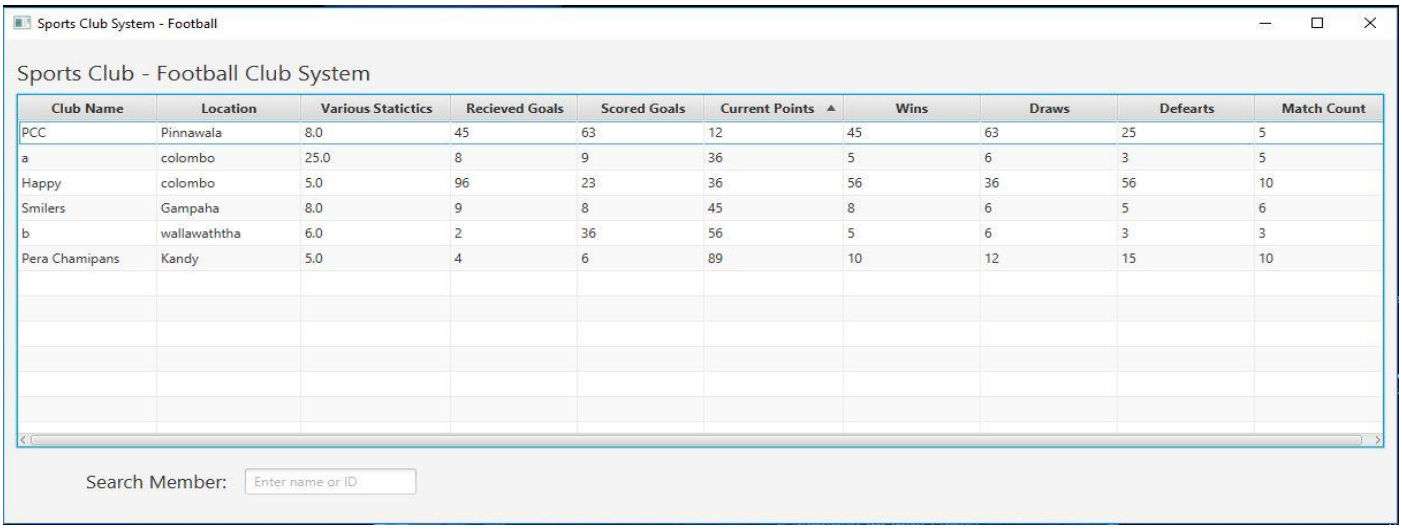
5.2 Filter Table



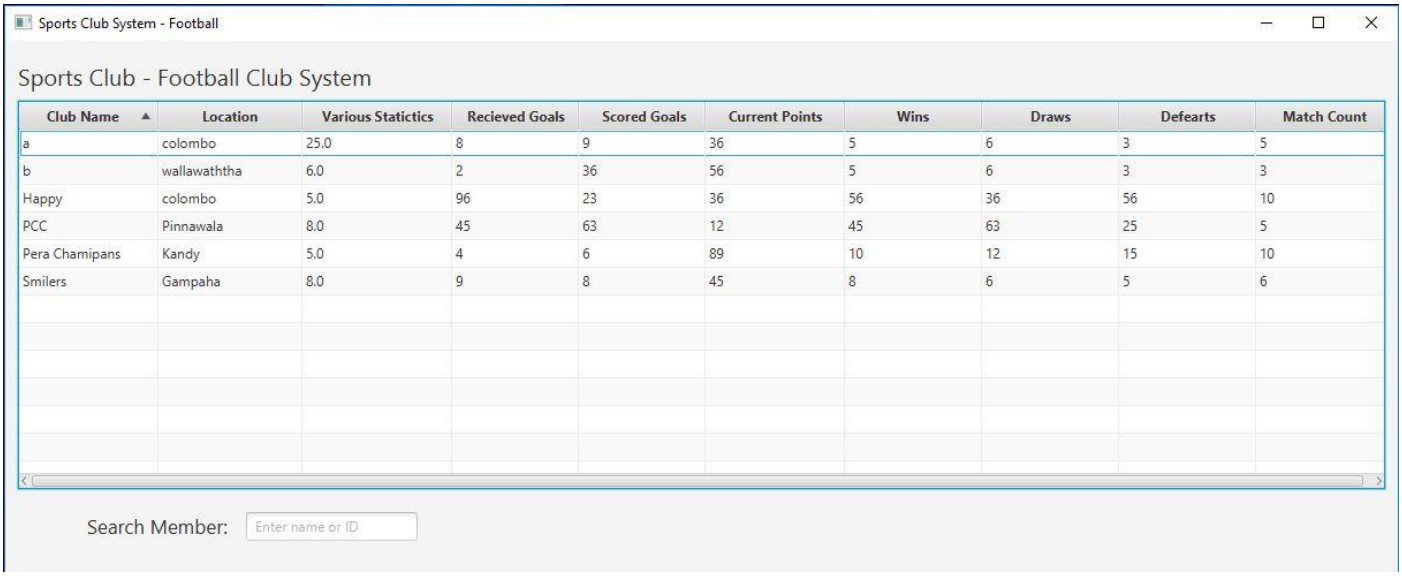


5.3 Sort Table

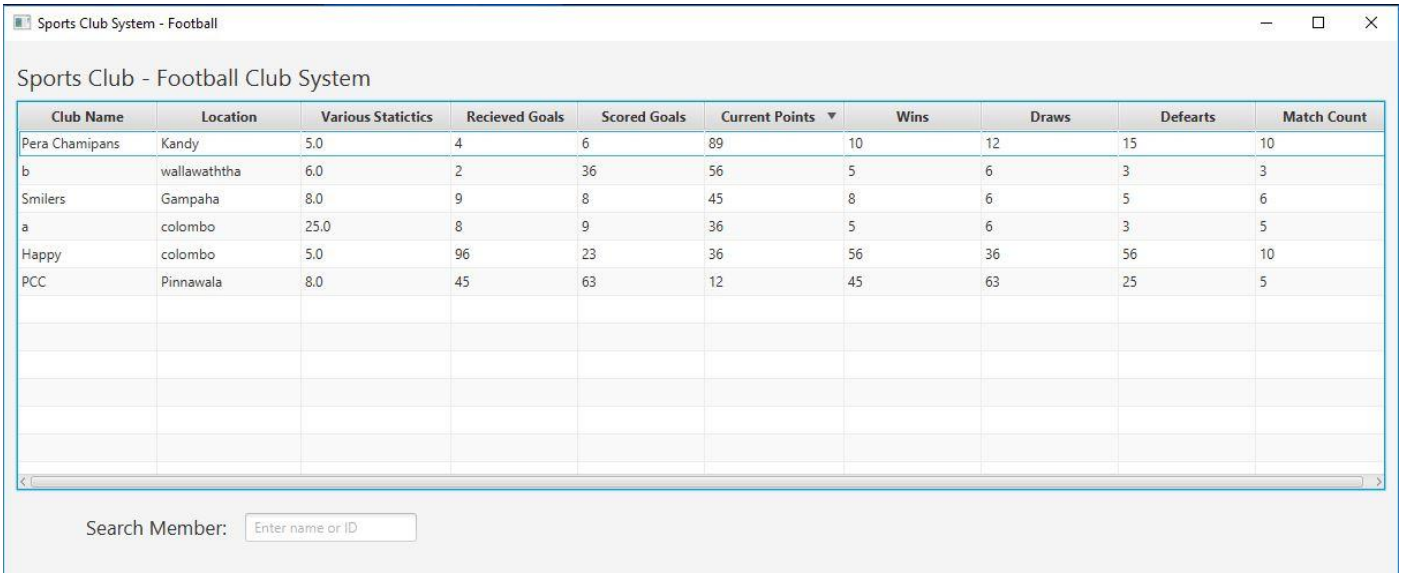
1. Assending order using current points



2. Alphabitical order using Sports Club name



3. Descending order using Current Points



5.4 GUI for Customer

6. Test Plan

Test Description	Expected Result	Actual Result
When user Input username and password	Display the console menu	Display the console menu
When user input incorrect user name and password	Display error message LOGIN UNSUCCESSFUL	Display error message LOGIN UNSUCCESSFUL
When the user select option number 1 from the menu to create sport club	The system will display list of questions and have 3 options to add sport club a - normal football club b - normal football club c - normal football club	The system will display list of questions and have 3 options to add sport club a - normal football club b - normal football club c - normal football club
When the user select option number 2 from the menu to delete a selected sport club	The system will ask the user to input name of the club user needs to be delete	The system will ask the user to input name of the club user needs to be delete
When the user select option number 3 from the menu to display a selected sport club	The system will ask the user to input name of the club user needs to be view and display the selected club	The system will ask the user to input name of the club user needs to be view and display the selected club
When the user select option number 4 from the menu to display all sport clubs	The system display the list of clubs in the sports club system	The system display the list of clubs in the sports club system
When the user select option number 5 from the menu to update a sport club	The system will ask the user to input name of the clubs needs to be updated and the points of the match	The system will ask the user to input name of the clubs needs to be updated and the points of the match
When the user select option number 6 from the menu to view table	This option will open the customer GUI in the browser	This option will open the customer GUI in the browser
When the user select option number 7 from the menu to Logout	The user will be logged out of the system SYSTEM IS SHUTTING DOWN	The user will be logged out of the system SYSTEM IS SHUTTING DOWN

7. Conclusion

This coursework is about the system for sports clubs and the system can create football club under the sports club and the system have three main football clubs. There are School football club, University football club and the normal football club. On the other hand the system can delete, view and display the selected club and the view the list of the sports club database. This system using file handling to store data and therefore The system have Graphical user interface to view all the data and it can be sorting ascending order and the descending order an alphabetical order therefore the GUI have search option and it is easy to sort club using there club names.

-THANK YOU-