



GROUP ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT038-3.5-2-ODJ

OBJECT-ORIENTED DEVELOPMENT WITH JAVA

HAND OUT DATE : 22 MARCH 2021

HAND IN DATE : 28 MAY 2021

WEIGHTAGE : 50%

Name	TP Number	Intake Code
LOKE WENG KHAY		UC2F2102CS(DA)

INSTRUCTIONS TO CANDIDATES:

- 1 Submit your assignment at the administrative counter**
- 2 Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing)**
- 3 Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld**
- 4 Cases of plagiarism will be penalized**
- 5 The assignment should be bound in an appropriate style (comb bound or stapled).**
- 6 Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.**
- 7 You must obtain 50% overall to pass this module.**

Table of Contents

I. Workload Distribution	3
1. Introduction.....	4
2. Assumptions.....	5
3. Design Diagrams	6
3.1 Use Case Diagram	6
3.2 Class Diagram	7
4. Object-Oriented Programming Concept	9
a. Class	9
b. Object.....	9
c. Encapsulation	10
d. Inheritance.....	12
e. Polymorphism	18
f. Aggregation	21
5. Screenshot of Output	22
6. Conclusion	93
References.....	94

I. Workload Distribution

No.	Task Component	Loke Weng Khay (TP062166)
1	Documentation	100%
2	Program Coding	100%

1. Introduction

Object-oriented programming has been a strong influence in the programming world as it is being used in education and the software development industry (Kölling, 1999). Object-Oriented Programming is a technique of creating objects that contains both data and function. Object-Oriented Programming is a suitable method of writing program codes as it allows for faster execution, clear structure of coding and reusability of the functions in other classes. Although Object Oriented Programming is good, not all of the programming languages support Object-Oriented Programming.

Object-Oriented Programming was coined by Alan Key in 1966 or 1967 when he was in grad school (Elliott, 2018). Ivan Sutherland's ground-breaking Sketchpad application was an early application for OOP, and it was created between 1961 and 1962 (Elliott, 2018). The first widely recognized "object-oriented" programming language was Simula, which was specified in 1965. Simula was similar to the sketchpad. Still, slowly and slowly, many other crucial features in object-oriented programming like inheritance, subclass, and many more have been added over time. (Elliott, 2018)

This report will describe the object-oriented programming used in the development of the Real Champions Sport Academy System.

2. Assumptions

This report will explain the assumption of this program, which is the Real Champions Sport Academy System. Sport Center HR Manager has approached us to assist them in developing the system.

This aims to develop a Real Champions Sport Academy system with three types of users: administrative, registered student and unregistered student. This project also aims to develop a database to store and archive all record that has been added or modified to the text files.

In this system, we have the admin role where they are able to manage and handle all coach, student, and sports records in the Real Champions Sport Academy System. The admin will have to log into the Real Champions Sport Academy System before the admin is able to make any changes. In the admin role, there are able to add new records for coaches, student or sports. Furthermore, they are able also to sort, display and search record for coaches, students or sports. Last and not least, the admin is able to modify any record in the system if any problem occurred. Any modification has to be updated in the text files as soon as the modification is completed.

There is also the registered student, where they can log into the system. Once in the system, the registered student is able to view their sports coach, self-records and registered sports schedule. Registered students are only able to add sport schedule based on the selected sport and provide feedback to the sports coaches based on their sports.

Last but not least, there are non-registered students, where they are not required to log in to the system. The only functionality is able to view all available sport and the sports schedule based on the available sport in the system. The non-registered student can also sign up to log in to the Real Champions Sport Academy System.

3. Design Diagrams

3.1 Use Case Diagram

Use case diagram is one of the subsets on unified modelling languages. The purpose of the use case diagram is to get the core functions of a system and envision the interactions between the actor and the Real Champions Sport Academy System. The figure below is the Use Case diagram for Real Champions Sport Academy System.

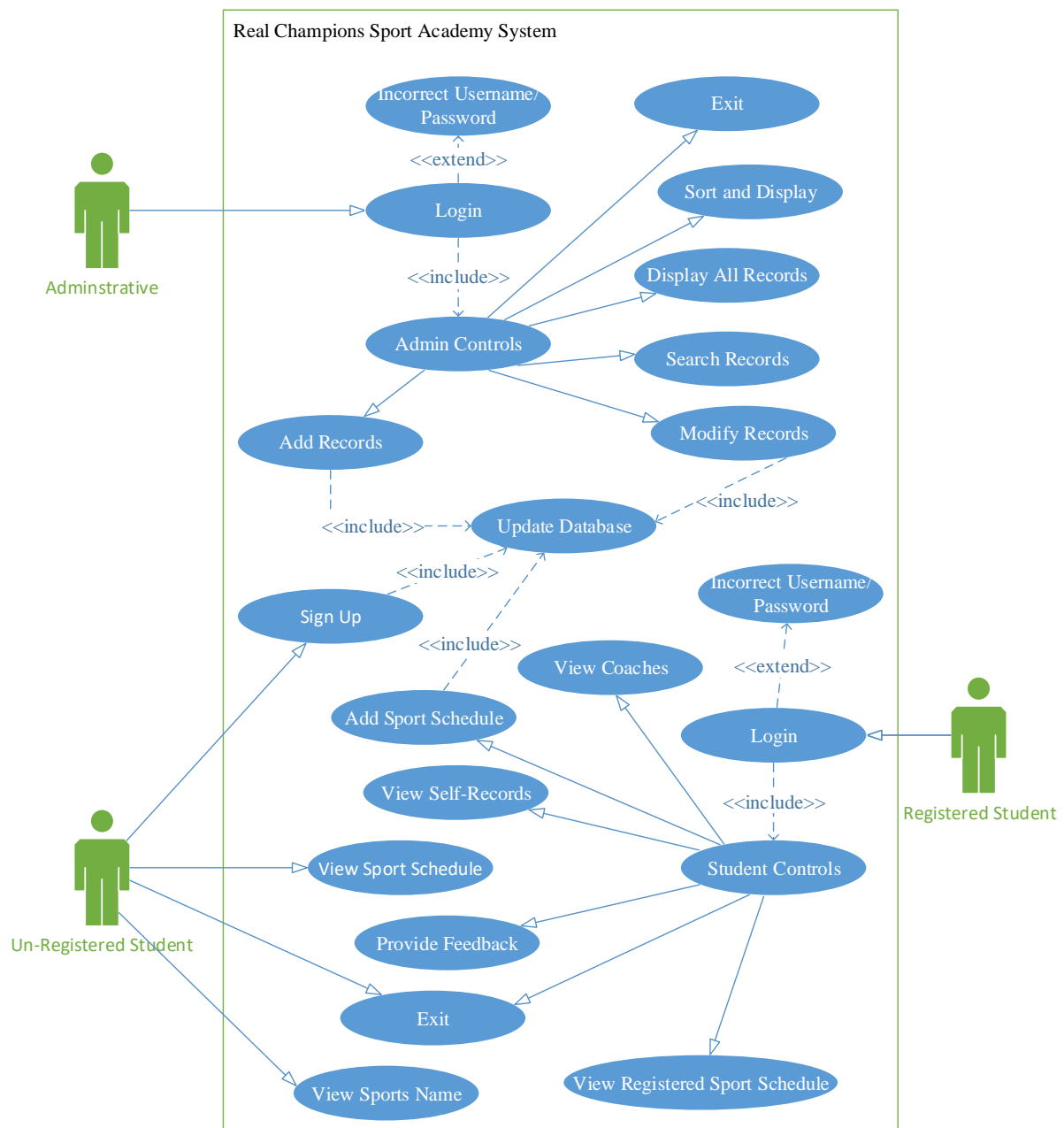


Figure 1: Use Case Diagram for Real Champions Sport Academy System

3.2 Class Diagram

Class diagrams are the primary building block of object-oriented modelling. They are used to represent the various objects in a system, their attributes, their operations, and the relationships between them (Nishadha, 2020). The figure below is the class diagram for the Real Champions Sport Academy System.

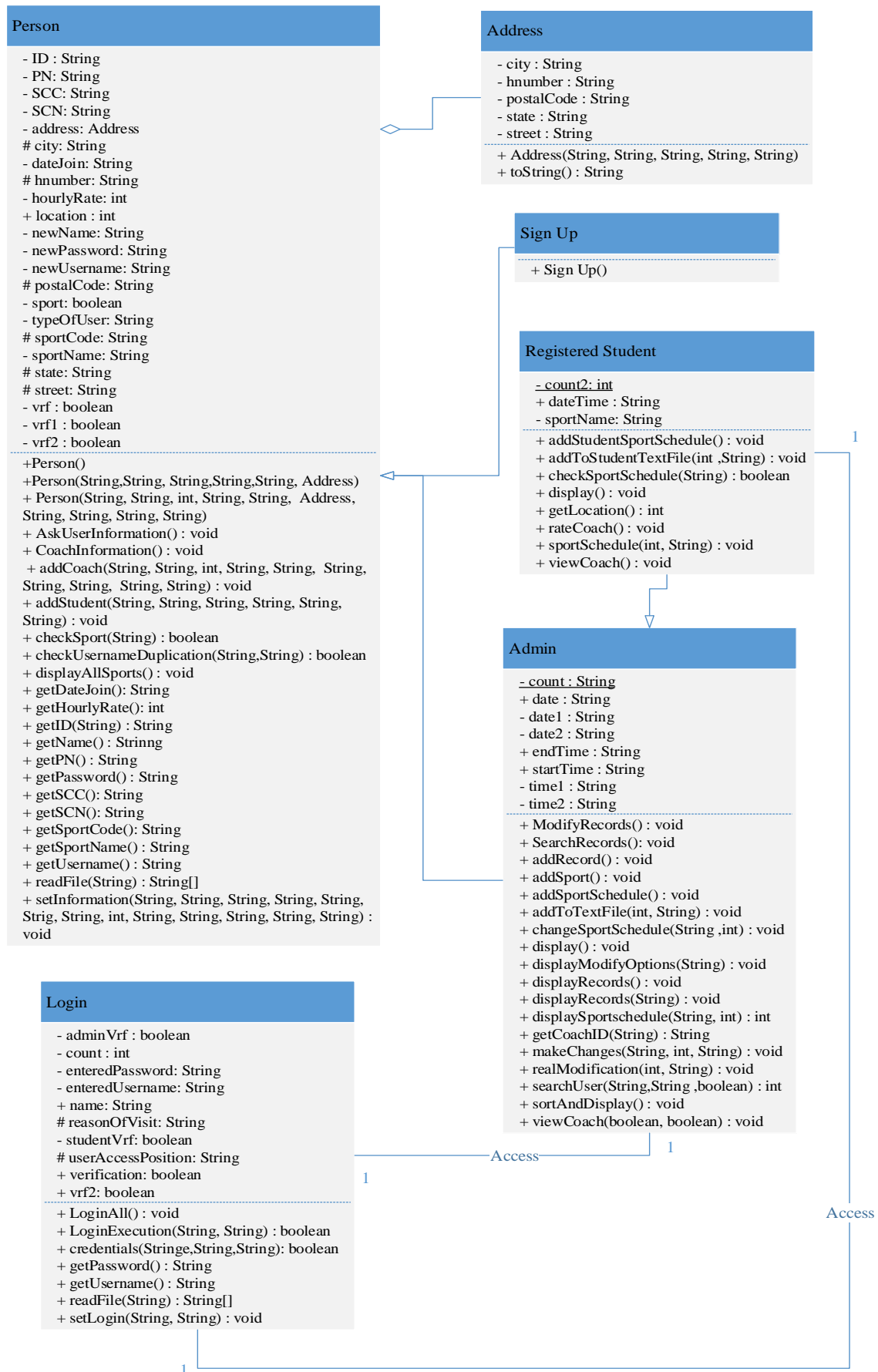


Figure 2: Class Diagram for Real Champions Sport Academy System

4. Object-Oriented Programming Concept

In this assignment, we were able to apply the object-oriented concept, which is Class. Object, Aggregation, Encapsulation, Polymorphism and Inheritance (Anon., n.d.). Below will describe what each concept is about and some example segment code from the project.

a. Class

In object-oriented programming, classes are the most basic concepts. A class describes the content that the object belongs to and describes the variable and methods in the class.

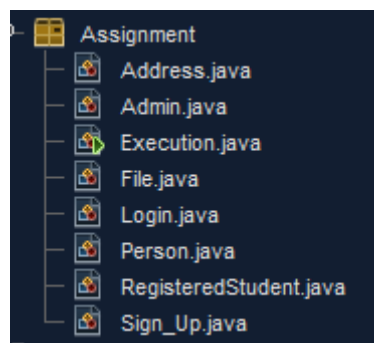


Figure 3: Assignment package has more than one classes

b. Object

An object is also crucial as they represent the identity of a class. The figure below will be used to give an overview of what an object is.

```
public class Sign_Up extends Person
{
    Sign_Up ()
    {
        AskUserInformation();
        Address address = new Address(hnumber, street, city, postalCode, state);
        new Person(getName(), getUsername(), getPassword(), "Student", getSportName(), address);
    }
}
```

Figure 4: Object used in “Sign_Up” Class

c. Encapsulation

Encapsulation is the best method to wrap code and data, and this is because encapsulation is known to secure data and prevent unauthorized changes to the data assigned. In figure 5, you could see that most of the data is set as private to avoid unauthorized changes to the initialize data. The data can only be edited using the setter method like in figure 6. If other classes want to view the data, they can request the getter method like in figure 7. With this, it allows for better data integrity of the system.

```
public class Person
{
    Scanner input = new Scanner(System.in);
    File rf4 = new File();
    private String newName;
    private String newUsername;
    private String newPassword;
    private String sportName;
    private String dateJoin;
    private boolean vrf=false,vrfl=false,vrf2=false,sport=false;
    private Address address;

    private String typeOfUser;
    private String dateTerminated;
    private int hourlyRate;
    private String SCC;
    private String SCN;
    protected String sportCode;
    private String ID;

    protected String hnumber;
    protected String street;
    protected String city;
    protected String state;
    protected String postalCode;
    private String PN;
```

Figure 5: The use of Encapsulation method in “Person” class (1)

```

/*Fuction: set data */
public void setInformation(String newName,String newUsername,String newPassword,String typeOfUser, Address address,String dateJoin,
    String dateTerminated,int hourlyRate,String SCC,String SCN,String sportCode,String sportName,String PN)
{
    this.newName=newName;
    this.newUsername=newUsername;
    this.newPassword=newPassword;
    this.typeOfUser=typeOfUser;
    this.address = address;
    this.dateJoin=dateJoin;
    this.dateTerminated=dateTerminated;
    this.hourlyRate=hourlyRate;
    this.SCC=SCC;
    this.SCN=SCN;
    this.sportName=sportName;
    this.sportCode=sportCode;
    this.PN=PN;
}

```

Figure 6: The use of Encapsulation method in “Person” class (2)

```

/*Fuction: Return Name*/
public String getName()
{
    return newName;
}
/*Fuction: Return Username*/
public String getUsername()
{
    return newUsername;
}
/*Fuction: Return Password*/
public String getPassword()
{
    return newPassword;
}
/*Fuction: Return Date Joined*/
public String getDateJoin()
{
    return dateJoin;
}
/*Fuction: Return Hourly Pay*/
public int getHourlyRate()
{
    return hourlyRate;
}

```

Figure 7: The use of Encapsulation method in “Person” class (3)

d. Inheritance

Inheritance is the ability to obtain properties from another class. It is like a parent-child relationship between two classes. The child class can acquire the parent class properties and methods when the “extends” keyword is inserted in the child class. The figure below is an example of how Inheritance works. Three types of Inheritance exist in java, which are Single Inheritance, Multilevel Inheritance and Hierarchical Inheritance (Anon., n.d.)

I. Single Inheritance

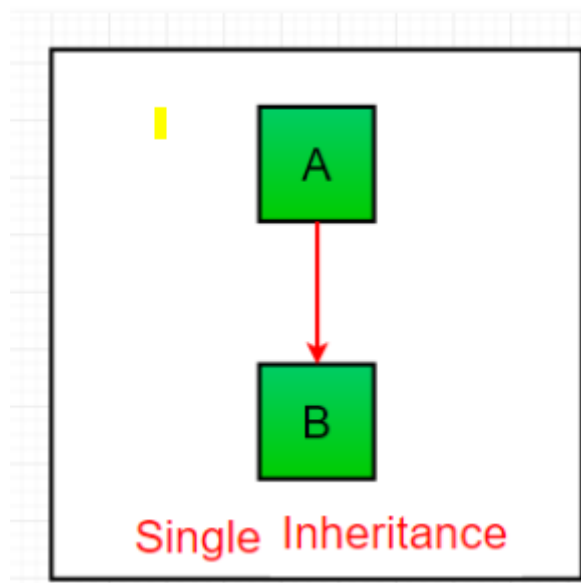


Figure 8: Single Inheritance Concept (Anon., 2021)

An example of single inheritance is Admin class (Parents Class) and RegisteredStudent class (Derived class).

```
public class Admin extends Person
{
    File rf2 = new File();
    Login log3 = new Login();
    Scanner input4= new Scanner(System.in);
    private String date1,date2,time1,time2;
    public static int count=0;
```

Figure 9: “Admin” Class (Base Class)

```
public class RegisteredStudent extends Admin
{
    Login log4 = new Login();
    File rfl = new File();
    Scanner input5= new Scanner(System.in);
    public static int count2=0;
    public String sportName;
```

Figure 10: “RegisteredStudent” Class (Derived Class)

II. Hierarchical Inheritance

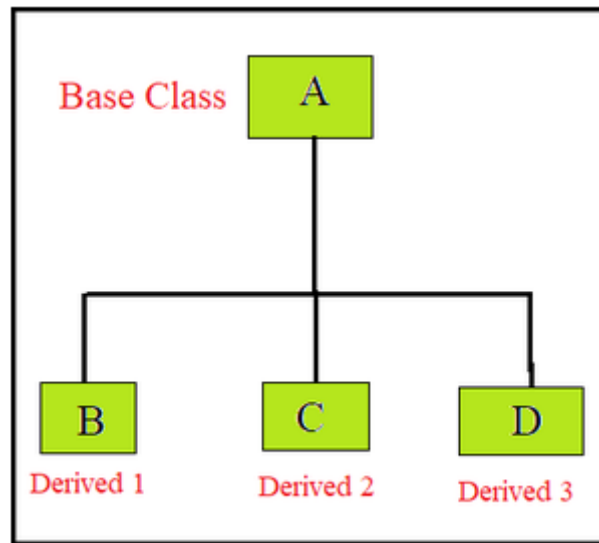


Figure 11: Hierarchical Inheritance Concept (Anon., 2021)

An example of hierarchical inheritance is the “Person” class (Parents) and “Sign_Up” class (Derived 1), and “Admin” class (Derived 2).

```
public class Person
{
    Scanner input = new Scanner(System.in);
    File rf4 = new File();
    private String newName;
    private String password;
```

Figure 12: Declaration of “Person” class (Base Class)

Once the “Person” class have been created, the “Sign_Up” class and “Admin” class can access the class by using the keyword “extends” as shown in figure 13 and figure 14. This concept is called Hierarchical Inheritance.

```

public class Sign_Up extends Person
{
    Sign_Up()
    {
        AskUserInformation();
        Address address = new Address(hnumber,street, city,postalCode,state);
        new Person(getName(),getUsername(),getPassword(),"Student",getSportName(),address);
    }
}

```

Figure 13: “Sign_Up” Class is using Inheritance to access “Person” class (Derived Class 1)

```

public class Admin extends Person
{
    File rf2 = new File();
    Login log3 = new Login();
    Scanner input4= new Scanner(System.in);
    private String date1,date2,time1,time2;
    public static int count=0;
}

```

Figure 14: “Admin” Class is using Inheritance to access “Person” class (Derived Class 2)

III. Multilevel Inheritance

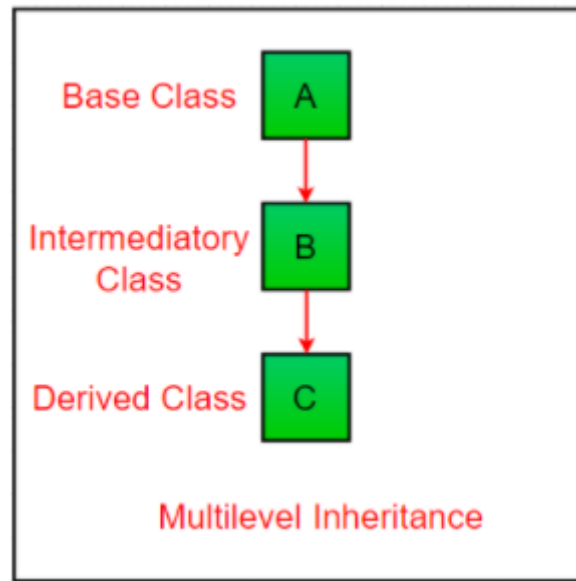


Figure 15: Multilevel Inheritance Concept (Anon., 2021)

```
public class Person
{
    Scanner input = new Scanner(System.in);
    File rf4 = new File();
    private String newName;
    private String email;
```

Figure 16: Declaration of “Person” class (Base Class)

```
public class Admin extends Person
{
    File rf2 = new File();
    Login log3 = new Login();
    Scanner input4= new Scanner(System.in);
    private String datel,date2,time1,time2;
    public static int count=0;
```

Figure 17: “Admin” Class is using Inheritance to access “Person” class (Intermediatory Class)


```
public class RegisteredStudent extends Admin
{
    Login log4 = new Login();
    File rfl = new File();
    Scanner input5= new Scanner(System.in);
}
```

Figure 18: “RegisteredStudent” Class is using Inheritance to access “Admin” class (Derived Class)

Once the “Person” class and “Admin” class is connected using the “extends” function, the “Registered Student” class can access both classes by using the keyword “extends” as shown in figure 18. This means that the “RegisteredStudent” class can access the “Person” class functions, but it has to go through the “Admin” class to access it. This concept is called Multilevel Inheritance.

e. Polymorphism

In polymorphism, it defines the ability of an object, variable or function to take multiple forms. In polymorphism, there two types which are method overloading and method overloading.

I. Method Overloading

In method overloading, usually, the methods will have the same name but different parameter. The figure below will describe two types of overloading which are method overloading and constructor overloading.

```
/*Basic Constructor */
Person()
{
}
/*Pre-defined Overloading Constructor */
Person(String newName,String newUsername,String newPassword
    ,String typeOfUser,String sportName, Address address)
{
    this.newName=newName;
    this.newUsername=newUsername;
    this.newPassword=newPassword;
    this.sportName=sportName;
    this.address = address;
    this.typeOfUser=typeOfUser;
    addStudent(getName(),getUsername(),getPassword(),address.toString(),"Student",
        getSportName());
}
/*Pre-defined Overloading Constructor */
Person(String newName,String dateJoin,int hourlyRate,String PN,String typeOfUser,
    Address address,String SCC,String SCN,String sportCode,String sportName)
{
    this.newName=newName;
    this.typeOfUser=typeOfUser;
    this.address = address;
    this.dateJoin=dateJoin;
    this.hourlyRate=hourlyRate;
    this.SCC=SCC;
    this.SCN=SCN;
    this.sportName=sportName;
    this.PN=PN;
    this.sportCode=sportCode;
    addCoach(getName(),getDateJoin(),getHourlyRate(),getPN(),typeOfUser,
        address.toString(),getSCC(),getSCN(),getSportCode(),getSportName());
}
```

Figure 19: Constructor Overloading in “Person” Class

```

public void displayRecords()
{
    String toView;
    boolean input2=false;
    System.out.println("What records would you like to display\n1. Coach\n2. Sport\n3. Registered Student");
    System.out.printf("Your Selection:");
    toView=input4.nextLine();
    System.out.println();
    while(input2 != true)
    {
        if(toView.equals("1") || toView.compareToIgnoreCase("Coach")==0)
        {
            displayRecords("Coach");
            input2 = true;
        }
        else if(toView.equals("2") || toView.compareToIgnoreCase("Sport")==0)
        {
            displayRecords("Sport");
            input2=true;
        }
        else if(toView.equals("3") || toView.compareToIgnoreCase("Student")==0)
        {
            displayRecords("Student");
            input2=true;
        }
        else
        {
            System.out.println("Please enter the right response");
        }
    }
}

//public String filename;
public void displayRecords(String filename)
{
    String[] line= new String[1000];
    int count=0;
    line=rf2.readFile(filename);
    System.out.println();
    while(line[count]!=null)
    {
        System.out.println(line[count]);
        count++;
    }
    System.out.println();
}

```

Figure 20: Method Overloading for “displayRecords” method in “Admin” Class

II. Method Overriding

In method overriding, the methods will usually have the same name and parameter, but each method is located in a different class. The figure below is an example of method overriding, which will describe the overriding method.

```
public class Admin extends Person
{
    File rf2 = new File();
    Login log3 = new Login();
    Scanner input4= new Scanner(System.in);
    private String datel,date2,timel,time2;
    public static int count=0;
    public void display()
    {
```

Figure 21: Method Overriding for “display” method in Admin Class (1)

```
public class RegisteredStudent extends Admin
{
    Login log4 = new Login();
    File rfl = new File();
    Scanner input5= new Scanner(System.in);
    public static int count2=0;
    public String sportName;
    @Override
    public void display()
    {
```

Figure 22: Method Overriding for “display” method in “RegisteredStudent” Class (2)

f. Aggregation

Aggregation is a relationship between two classes where their relationship is more to “has a” relationship or “whole/part” of relationship. The aggregation class usually is a reference to another class, and it has ownership of the class they have been aggregated with (Leahy, 2019). Any class that has been referenced are considered as a part of the aggregation class. The figure below will describe the application of the aggregation class in the assignment.

```
public class Person
{
    Scanner input = new Scanner(System.in);
    File rf4 = new File();
    private String newName;
    private String newUsername;
    private String newPassword;
    private String sportName;
    private String dateJoin;
    private boolean vrf=false,vrfl=false,vrf2=false,sport=false;
    private Address address;
```

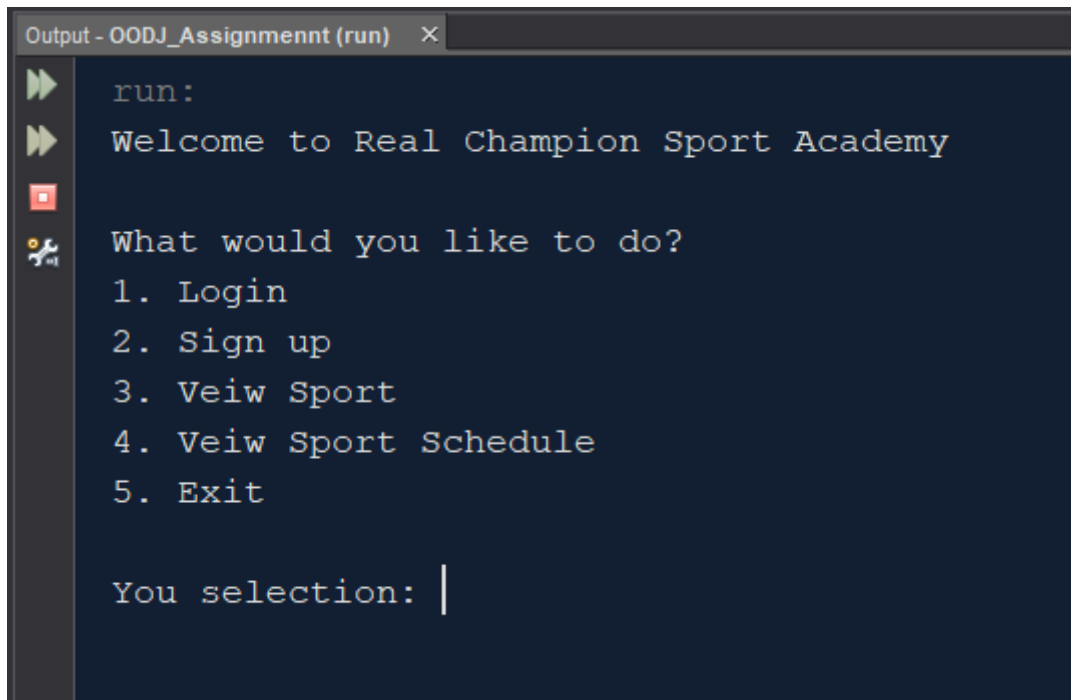
Figure 23: Declare the use of aggregation for “Address” class (Aggregation Class) in “Person” class (Base Class)

```
public class Address
{
    Scanner input = new Scanner(System.in);
    private String hnumber;
    private String street;
    private String city;
    private String state;
    private String postalCode;
    Address(String hNumber,String street, String city, String postalCode,String state)
    {
        this.hnumber=hNumber;
        this.street=street;
        this.city=city;
        this.postalCode=postalCode;
        this.state=state;
    }
    public String toString()
    {
        return "No. " + hnumber + ", " + street + ", " + city + ", " + postalCode + ", " + state;
    }
}
```

Figure 24: “Address” class (Aggregation Class)

5. Screenshot of Output

The figure below will describe how the system will look like.



```
Output - OODJ_Assignmennt (run) ×
run:
Welcome to Real Champion Sport Academy
What would you like to do?
1. Login
2. Sign up
3. Veiw Sport
4. Veiw Sport Schedule
5. Exit
You selection: |
```

Figure 25: Greeting Page in Real Champion Sport Academy System

User will be greeted by the Real Champion Sport Academy System. The system will display the front page displaying the option the user will want to do. There are five functions the user can do on the front page, which are:

- Login
- Sign Up
- View All Sports
- View Sport Schedule
- Exit Real Champion Sport Academy System

The following figure will describe each function.

1. Login Function

In this system, there are two types of account in the Real Champion Sport Academy System which are:

- Administrative Account
- Registered Student Account

The user can log in as an admin or a student based on their user access. We will first describe the Administrative Account Functions.

Administrative Account

Once the user has login as an Administrative, there will be greeted with functions that allow them to manage all coaches, students and sports. The figure below shows the functions available in the Administrative Function

```
You selection: 1
You selected: Login

Enter username
Justsomalostguy
Enter password
123

Welcome, Loke Weng Khay

What would you like to do today
1. Add Record
2. Display All Records
3. Search Specific Record
4. Sort and Display Record
5. Modify Existing Record
6. Exit
Your Selection: |
```

Figure 26: Administrative Account in the Real Champion Sport Academy System

Administrative Account Functions:

1. Add Record Function

In the add record function, administrative are able to add new coach, student, sport and sport schedule record to the Real Champion Sport Academy System.

```
Welcome, Loke Weng Khay

What would you like to do today
1. Add Record
2. Display All Records
3. Search Specific Record
4. Sort and Display Record
5. Modify Existing Record
6. Exit
Your Selection: add record
What record would you like to add?
1. Coach
2. Student
3. Sport
4. Sport Schedule
Your Selection:|
```

Figure 27: Options when using “Add Record” Function

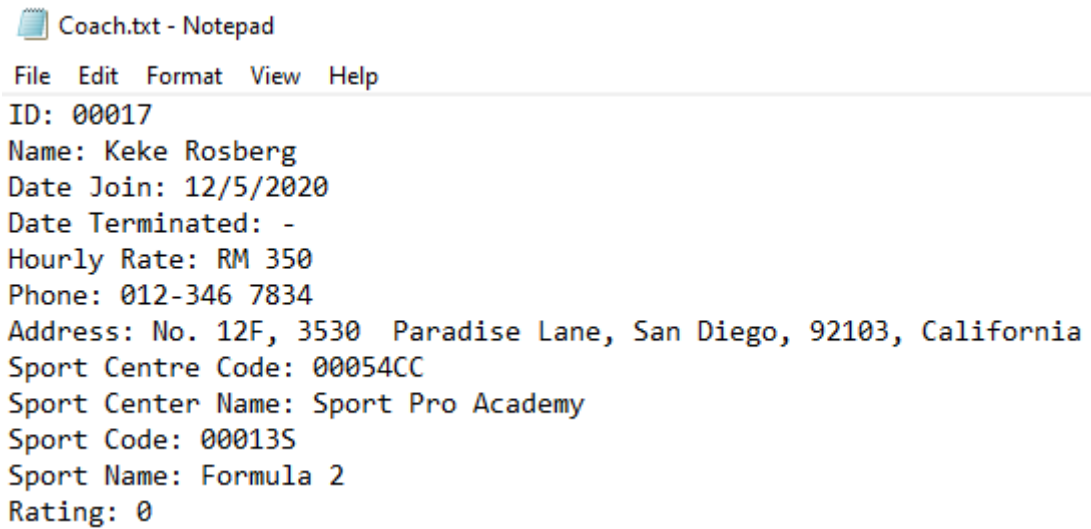
1.1 Add Coach

The figure below is an example of adding new coach data into the Real Champion Sport Academy System.

```
What record would you like to add?
1. Coach
2. Student
3. Sport
4. Sport Schedule
Your Selection: Coach

Coach Information:
Enter Full Name
Keke Rosberg
Enter Join Date: dd/mm/yyyy
12/5/2020
Please enter hourly pay rate
RM 350
Enter phone number
012-346 7834
Enter house number
No. 12F
Enter street
3530 Paradise Lane
Enter City
San Diego
Enter Postal Code
92103
Enter State
California
Enter Sport Centre Code
00054CC
Enter Sport Center Name
Sport Pro Academy
Enter Sport Name
Formula 2
```

Figure 28: Adding new Coach



Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Keke Rosberg
Date Join: 12/5/2020
Date Terminated: -
Hourly Rate: RM 350
Phone: 012-346 7834
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 000135
Sport Name: Formula 2
Rating: 0

Figure 29: New coach data inserted into “Coach.txt” file after code execution

After adding all the information, the coach’s data will be saved into the “Coach.txt” file. This allows the data to be saved to the Real Champion Sport Academy System.

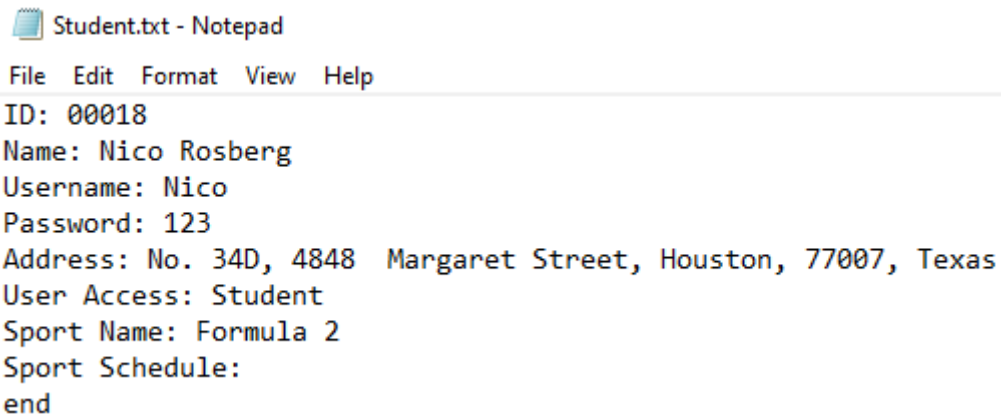
1.2 Add Student

The figure below is an example of adding new student data into the Real Champion Sport Academy System.

```
What record would you like to add?
1. Coach
2. Student
3. Sport
4. Sport Schedule
Your Selection: Student

Student Information:
Enter Full Name
Nico Rosberg
Enter Username:
Nico
Enter Password:
123
Re-Enter Password again
123
Enter Sport Name
Formula 2
Enter house number
No. 34D
Enter street
4848 Margaret Street
Enter City
Houston
Enter Postal Code
77007
Enter State
Texas
```

Figure 30: Adding new Student



Student.txt - Notepad

File Edit Format View Help

ID: 00018
Name: Nico Rosberg
Username: Nico
Password: 123
Address: No. 34D, 4848 Margaret Street, Houston, 77007, Texas
User Access: Student
Sport Name: Formula 2
Sport Schedule:
end

Figure 31: New student data inserted into “Student.txt” file after code execution

After adding all the information, the student’s data will be saved into the “Student.txt” file. This allows the data to be saved to the Real Champion Sport Academy System.

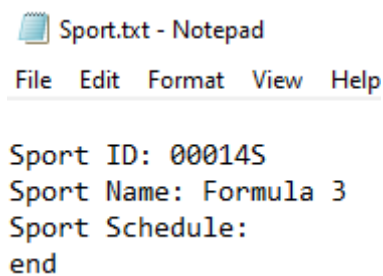
1.3 Add Sport

The figure below is an example of adding sport data into the Real Champion Sport Academy System.

```
What record would you like to add?
1. Coach
2. Student
3. Sport
4. Sport Schedule
Your Selection: Sport

Enter new sport
Formula 3
```

Figure 32: Adding new Sport



Sport.txt - Notepad

File Edit Format View Help

Sport ID: 000145
Sport Name: Formula 3
Sport Schedule:
end

Figure 33: New sport data inserted into “Sport.txt” file after code execution

After adding all the information, the sport’s data will be saved into the “Sport.txt” file. This allows the data to be saved to the Real Champion Sport Academy System.

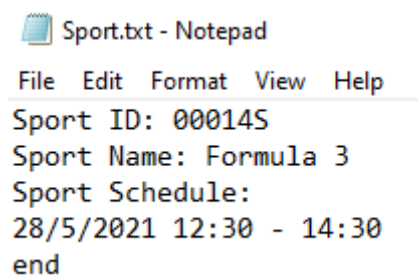
1.4 Add Sport Schedule

The figure below is an example of adding sport schedule data into the Real Champion Sport Academy System.

```
What record would you like to add?
1. Coach
2. Student
3. Sport
4. Sport Schedule
Your Selection: Sport Schedule

Enter sport name
Formula 3
Enter Date for Formula 3 class: dd/mm/yyyy
28/5/2021
Enter start time for the class: hh:ss (24-hours)
12:30
Enter end time for the class: hh:ss (24-hours)
14:30
Successfully wrote to the file.
```

Figure 34: Adding new Sport Schedule



```
Sport ID: 000145
Sport Name: Formula 3
Sport Schedule:
28/5/2021 12:30 - 14:30
end
```

Figure 35: New sport schedule data inserted into “Sport.txt” file after code execution

After adding all the information, the sport schedule data will be saved into the “Sport.txt” file. This allows the data to be saved to the Real Champion Sport Academy System.

2. Display All Records Function

```
What would you like to do today
1. Add Record
2. Display All Records
3. Search Specific Record
4. Sort and Display Record
5. Modify Existing Record
6. Exit
Your Selection: 2
What records would you like to display
1. Coach
2. Sport
3. Registered Student
Your Selection: |
```

Figure 36: All options for Display All function

The figure above shows all the available option to display all records. The administrative can display all coach, student and sports record.

2.1 Display All Coach Function

```
What records would you like to display
1. Coach
2. Sport
3. Registered Student
Your Selection: Coach

ID: 00002
Name: Joseph Dan
Date Joined: 23/2/2020
Date Terminated: 12/3/2021
Hourly Rate: RM 200
Phone: 012-345 6789
Address: No.123, 493 Brookview Drive, Smith Point, 77514, Texas
Sport Centre Code: 00001CC
Sport Center Name: Sport Centre 1
Sport Code: 00001S
Sport Name: Swimming
Rating: 0

ID: 00005
Name: Emily Elledge
Date Joined: 20/11/2020
Date Terminated: -
Hourly Rate: RM 150
Phone: 012-482 6754
Address: No.67, 4617 School Street, LOTT, 76656, Texas
Sport Centre Code: 00021CC
Sport Center Name: Sport Pros
Sport Code: 00002S
Sport Name: Badminton
Rating: 4
```

Figure 37: All Coach Records is being displayed

Figure 37 represents all coach data being displayed to the administrative user.

2.2 Display All Sport Function

```
What records would you like to display
1. Coach
2. Sport
3. Registered Student
Your Selection: Sport

Sport ID: 00001S
Sport Name: Swimming
Sport Schedule:
end

Sport ID: 00002S
Sport Name: Badminton
Sport Schedule:
15/1/2020 19:00 - 23:00
19/1/2020 12:00 - 14:00
end
```

Figure 38: All Sport Records is being displayed

Figure 38 represents all sport data being displayed to the administrative user.

2.3 Display All Registered Student

```
What records would you like to display
1. Coach
2. Sport
3. Registered Student
Your Selection: Student

ID: 00003
Name: Sebestian Vettel
Username: Vettel
Password: 123
Address: No. 12, Taman Cantik, Bandar Baru, 12345, Selangor
User Access: Student
Sport Name: Badminton
Sport Schedule:
end

ID: 00004
Name: Alex Albon
Username: Alex
Password: Testing
Address: No. 1302, Jefferson Street, Portsmouth, 23707, Virginia
User Access: Student
Sport Name: Archery
Sport Schedule:
end
```

Figure 39: All Student Records is being displayed

Figure 39 represents all student data being displayed to the administrative user.

3. Search Specific Records Function

```
What would you like to do today
1. Add Record
2. Display All Records
3. Search Specific Record
4. Sort and Display Record
5. Modify Existing Record
6. Exit
Your Selection: 3
Who would you like to search:
1. Coach
2. Student
3. Sport
Your Selection: |
```

Figure 40: All options to search user

Figure 40 represents all the options to search for a particular user in the Real Champion Sport Academy System.

3.1.1 Search Coach by Coach ID

```
Who would you like to search:
1. Coach
2. Student
3. Sport
Your Selection: Coach

Search by:
1. Coach ID
2. Performance Rating
Your Selection:Coach ID

Enter Coach ID:
00017

ID: 00017
Name: Keke Rosberg
Date Join: 12/5/2020
Date Terminated: -
Hourly Rate: RM 350
Phone: 012-346 7834
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0
```

Figure 41: Search Coach using Coach ID

Figure 41 represents searching coach based on the coach ID. Once the coach is found, the program will display the coach's records.

3.1.2 Search Coach using Performance Rating

```
Who would you like to search:
1. Coach
2. Student
3. Sport
Your Selection: Coach

Search by:
1. Coach ID
2. Performance Rating
Your Selection:2

Enter Performance Rating (0-5)
4

ID: 00005
Name: Emily Elledge
Date Joined: 20/11/2020
Date Terminated: -
Hourly Rate: RM 150
Phone: 012-482 6754
Address: No.67, 4617 School Street, LOTT, 76656, Texas
Sport Centre Code: 00021CC
Sport Center Name: Sport Pros
Sport Code: 00002S
Sport Name: Badminton
Rating: 4
```

Figure 42: Search Coach using Performance Rating

Figure 42 represents searching coach based on their performance rating. Once the coach is found, the program will display the coach's records. If there are multiple coaches with the same performance rating, it will display all the coach with the same performance rating.

3.2 Search Student

```
Who would you like to search:
1. Coach
2. Student
3. Sport
Your Selection: Student

Enter Student ID
00016

ID: 00016
Name: Nick Latifi
Username: Nicholas
Password: Nick123
Address: 56G, 796 Still Street, Crestline, 44827, Ohio
User Access: Student
Sport Name: Formula 3
Sport Schedule:
15/5/2021 12:30 - 14:30
```

Figure 43: Search Student using Student ID

Figure 43 represents searching student based on the Student ID. Once the student is found, the program will display the student's records.

3.3 Search Sport

```
Who would you like to search:
1. Coach
2. Student
3. Sport
Your Selection: Sport

Enter Sport ID
00013S

Sport ID: 00013S
Sport Name: Formula 2
Sport Schedule:
23/5/2021 12:30 - 14:30
29/5/2021 14:30 - 16:30
```

Figure 44: Search Sport using Sport ID

Figure 44 represents searching sport based on the Sport ID. Once the sport is found, the program will display the sport's records.

4. Sort and Display Record Function

```
What would you like to do today
1. Add Record
2. Display All Records
3. Search Specific Record
4. Sort and Display Record
5. Modify Existing Record
6. Exit
Your Selection: 4

Sort and Display:
1. Coach in ascending order
2. Coach Hour Pay in ascending order
3. Coach Performance Rating in ascending order
Your Selection:
```

Figure 45: All Options available for Sort and Display Function

Figure 45 represent all the options available to sort and display coaches based on multiple conditions. The following figure will explain the differences between each of the options.

4.1 Coach in ascending order

```
Sort and Display:
1. Coach in ascending order
2. Coach Hour Pay in ascending order
3. Coach Performance Rating in ascending order
Your Selction: 1

-----
| ID | Name |
-----
|00005|Emily Elledge|
-----
|00002|Joseph Dan|
-----
|00017|Keke Rosberg|
-----
|00007|Pandora Howe|
-----
|00006|Queenie Lindsey|
-----
|00015|Sergio|
-----
|00013|Valteri Bottas|
-----
|00012|Zoe Mcdaniel|
-----
```

Figure 46: Sort and Display Coaches in ascending order

Based on figure 46 represent sorting and displaying the coaches in ascending order. This means that the program will sort the coaches based on their name in ascending order.

4.2 Coach Hour Pay in ascending order

```
Sort and Display:
1. Coach in ascending order
2. Coach Hour Pay in ascending order
3. Coach Performance Rating in ascending order
Your Selction: 2

-----
| ID |           Name           | Hourly Rate (RM) |
-----
|00005|Emily Elledge             |         150       |
-----
|00007|Pandora Howe              |         150       |
-----
|00006|Queenie Lindsey           |         190       |
-----
|00002|Joseph Dan                |         200       |
-----
|00012|Zoe Mcdaniel              |         280       |
-----
|00017|Keke Rosberg              |         350       |
-----
|00015|Sergio                    |         450       |
-----
|00013|Valteri Bottas            |         495       |
-----
```

Figure 47: Sort and Display Coaches based on Hourly Pay in ascending order

Figure 47 represents sorting and displaying the coaches based on their hourly pay in ascending order. This means that the program will sort the coaches based on their hourly pay in ascending order.

4.3 Coach Performance Rating in ascending order

```
Sort and Display:
1. Coach in ascending order
2. Coach Hour Pay in ascending order
3. Coach Performance Rating in ascending order
Your Selction: 3

-----
| ID |           Name           |Rating|
-----
|00002|Joseph Dan               |  0   |
-----
|00006|Queenie Lindsey          |  0   |
-----
|00007|Pandora Howe             |  0   |
-----
|00012|Zoe Mcdaniel             |  0   |
-----
|00013|Valteri Bottas           |  0   |
-----
|00017|Keke Rosberg             |  0   |
-----
|00005|Emily Elledge            |  4   |
-----
|00015|Sergio                   |  5   |
-----
```

Figure 48: Sort and Display Coaches based on Performance Rating in ascending order

Figure 48 represents sorting and displaying the coaches based on their performance rating in ascending order. This means that the program will sort the coaches based on their performance rating in ascending order.

5. Modify Function

```
What would you like to do today
1. Add Record
2. Display All Records
3. Search Specific Record
4. Sort and Display Record
5. Modify Existing Record
6. Exit
Your Selection: 5
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: |
```

Figure 49: All Modification Options for Modify Functions

Figure 49 represent the modification options for all user in the Real Champion Sport Academy System. The administration has access to modify existing records for coaches, students and sports.

5.1 Modify Coach Record

5.1.1 Name

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Coach

Please select on how you want to find the user:
1. ID
2. Fullname
Your Selection: ID


Enter ID:
00017

User Found
What would you like to modify
1. Name
2. Date Joined
3. Date Terminated
4. Hourly Rate
5. Phone Number
6. Address
7. Sport Centre Code
8. Sport Center Name
9. Sport Name
Your Selection: 1

Enter Name
Neke Rosberg
File have been updated
```

Figure 50: Modify Coach Name

Figure 50 represent the modification to the coach's name. The coach's name will be replaced with the new name that was entered by the administrative user. The changes will be updated to the text files immediately.




Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Keke Rosberg
Date Join: 12/5/2020
Date Terminated: -
Hourly Rate: RM 350
Phone: 012-346 7834
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0

Figure 51: Coach Name before modification in “Coach.txt” file



Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Join: 12/5/2020
Date Terminated: -
Hourly Rate: RM 350
Phone: 012-346 7834
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0

Figure 52: Coach Name after modification in “Coach.txt” file

5.1.2 Date Joined

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Coach

Please select on how you want to find the user:
1. ID
2. Fullname
Your Selection: 2

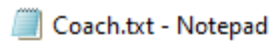
Enter Fullname:
Neke Rosberg

User Found
What would you like to modify
1. Name
2. Date Joined
3. Date Terminated
4. Hourly Rate
5. Phone Number
6. Address
7. Sport Centre Code
8. Sport Center Name
9. Sport Name
Your Selection: 2

Enter Date Joined: dd/mm/yyyy
25/5/2020
File have been updated
```

Figure 53: Modify Coach Date Joined

Figure 53 represent the modification to the coach's date joined. The coach's joined date will be replaced with the new joined date that was entered by the administrative user. The changes will be updated to the text files immediately.

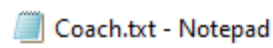


Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Join: 12/5/2020
Date Terminated: -
Hourly Rate: RM 350
Phone: 012-346 7834
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 000135
Sport Name: Formula 2
Rating: 0

Figure 54: Coach Date Join before modification in “Coach.txt” file



Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: -
Hourly Rate: RM 350
Phone: 012-346 7834
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 000135
Sport Name: Formula 2
Rating: 0

Figure 55: Coach Date Join after modification in “Coach.txt” file

5.1.3 Date Terminated

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Coach

Please select on how you want to find the user:
1. ID
2. Fullname
Your Selection: 1


Enter ID:
00017

User Found
What would you like to modify
1. Name
2. Date Joined
3. Date Terminated
4. Hourly Rate
5. Phone Number
6. Address
7. Sport Centre Code
8. Sport Center Name
9. Sport Name
Your Selection: 3

Enter Date Terminated: dd/mm/yyyy
30/5/2021
File have been updated
```

Figure 56: Modify Coach Date Terminated

Figure 56 represent the modification to the coach's termination date. The coach's termination date will be replaced with the new termination date that was entered by the administrative user. The changes will be updated to the text files immediately.

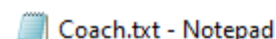


Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: -
Hourly Rate: RM 350
Phone: 012-346 7834
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 000135
Sport Name: Formula 2
Rating: 0

Figure 57: Coach Date Terminated before modification in “Coach.txt” file



Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 350
Phone: 012-346 7834
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 000135
Sport Name: Formula 2
Rating: 0

Figure 58: Coach Date Terminated after modification in “Coach.txt” file

5.1.4 Hourly Rate

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: 1

Please select on how you want to find the user:
1. ID
2. Fullname
Your Selection: 1


Enter ID:
00017

User Found
What would you like to modify
1. Name
2. Date Joined
3. Date Terminated
4. Hourly Rate
5. Phone Number
6. Address
7. Sport Centre Code
8. Sport Center Name
9. Sport Name
Your Selection: 4

Enter Hourly Rate
450
File have been updated
```

Figure 59: Modify Coach Hourly Rate

Figure 59 represent the modification to the coach's hourly rate pay. The coach's hourly rate pay will be replaced with the new hourly rate pay that was entered by the administrative user. The changes will be updated to the text files immediately.




Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 350
Phone: 012-346 7834
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 000135
Sport Name: Formula 2
Rating: 0

Figure 60: Coach Hourly Rate before modification in “Coach.txt” file



Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 450
Phone: 012-346 7834
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 000135
Sport Name: Formula 2
Rating: 0

Figure 61: Coach Hourly Rate after modification in “Coach.txt” file

5.1.5 Phone Number

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Coach

Please select on how you want to find the user:
1. ID
2. Fullname
Your Selection: 2


Enter Fullname:
Neke Rosberg

User Found
What would you like to modify
1. Name
2. Date Joined
3. Date Terminated
4. Hourly Rate
5. Phone Number
6. Address
7. Sport Centre Code
8. Sport Center Name
9. Sport Name
Your Selection: Phone Number

Enter Phone Number
012-345 6789
File have been updated
```

Figure 62: Modify Coach Phone Number


Figure 62 represent the modification to the coach's phone number. The coach's phone number will be replaced with the new phone number that was entered by the administrative user. The changes will be updated to the text files immediately.

 Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 450
Phone: 012-346 7834
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0

Figure 63: Coach Phone Number before modification in “Coach.txt” file

 Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 450
Phone: 012-345 6789
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0

Figure 64: Coach Phone Number after modification in “Coach.txt” file

5.1.6 Address

```
Please select on how you want to find the user:
1. ID
2. Fullname
Your Selection: 1

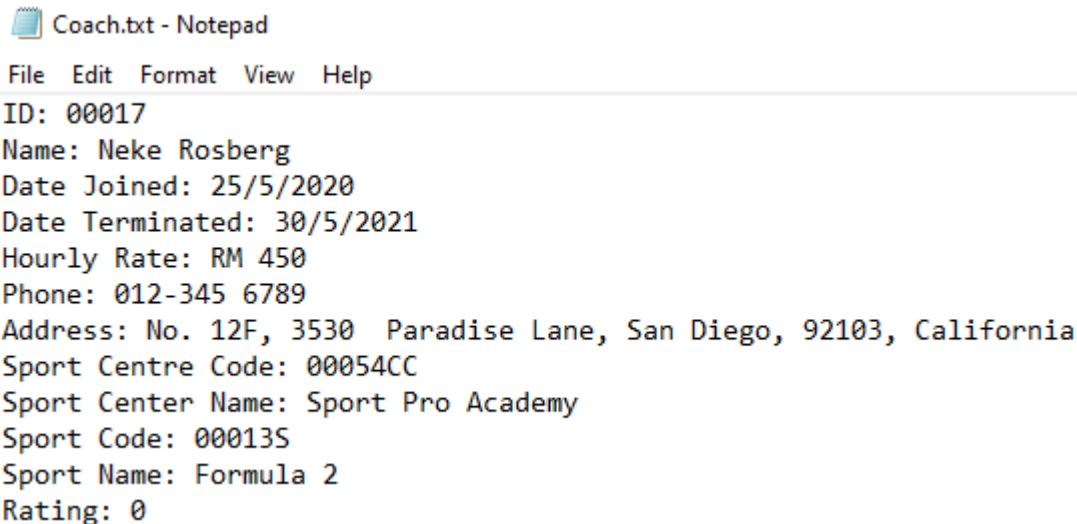
Enter ID:
00017

User Found
What would you like to modify
1. Name
2. Date Joined
3. Date Terminated
4. Hourly Rate
5. Phone Number
6. Address
7. Sport Centre Code
8. Sport Center Name
9. Sport Name
Your Selection: 6

Enter house number
No. 45G
Enter street
668 Cody Ridge Road
Enter City
Cleo Springs
Enter Postal Code
73729
Enter State
Oklahoma
File have been updated
```

Figure 65: Modify Coach Address

Figure 65 represent the modification to the coach's address. The coach's address will be replaced with the new address that was entered by the administrative user. The changes will be updated to the text files immediately.

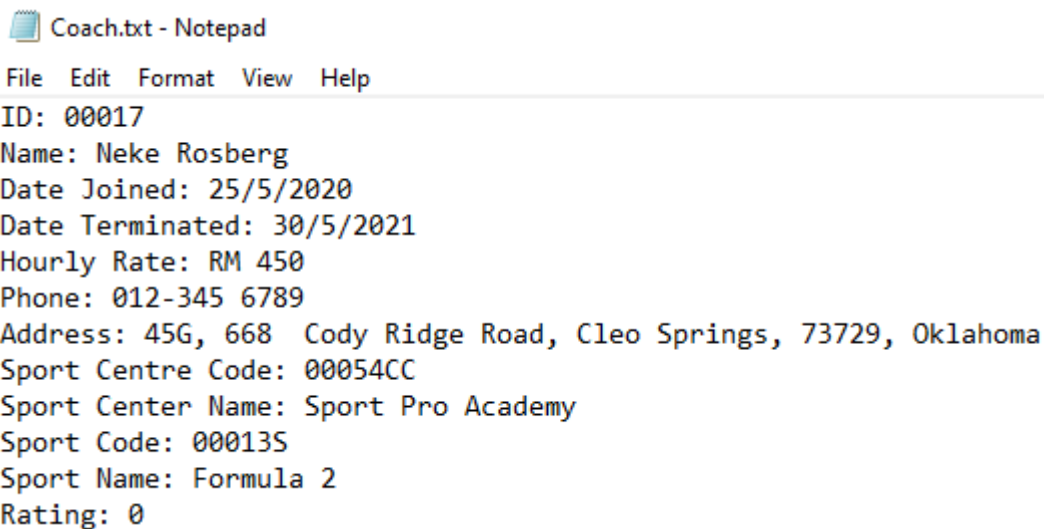


Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 450
Phone: 012-345 6789
Address: No. 12F, 3530 Paradise Lane, San Diego, 92103, California
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0

Figure 66: Coach Address before modification in “Coach.txt” file



Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 450
Phone: 012-345 6789
Address: 45G, 668 Cody Ridge Road, Cleo Springs, 73729, Oklahoma
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0

Figure 67: Coach Address after modification in “Coach.txt” file

5.1.7 Sport Centre Code

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Coach

Please select on how you want to find the user:
1. ID
2. Fullname
Your Selection: 2

Enter Fullname:
Neke Rosberg

User Found
What would you like to modify
1. Name
2. Date Joined
3. Date Terminated
4. Hourly Rate
5. Phone Number
6. Address
7. Sport Centre Code
8. Sport Center Name
9. Sport Name
Your Selection: 7

Enter Sport Centre Code
00010CC
File have been updated
```

Figure 68: Modify Coach Sport Center Code

Figure 68 represent the modification to the coach's sport center code. The coach's sport center code will be replaced with the new sport center code that was entered by the administrative user. The changes will be updated to the text files immediately.


 Coach.txt - Notepad
File Edit Format View Help
ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 450
Phone: 012-345 6789
Address: 45G, 668 Cody Ridge Road, Cleo Springs, 73729, Oklahoma
Sport Centre Code: 00054CC
Sport Center Name: Sport Pro Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0

Figure 69: Coach Sport Center Code before modification in “Coach.txt” file


 Coach.txt - Notepad
File Edit Format View Help
ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 450
Phone: 012-345 6789
Address: 45G, 668 Cody Ridge Road, Cleo Springs, 73729, Oklahoma
Sport Centre Code: 00010CC
Sport Center Name: Sport Pro Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0

Figure 70: Coach Sport Center Code after modification in “Coach.txt” file

5.1.8 Sport Center Name

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Coach

Please select on how you want to find the user:
1. ID
2. Fullname
Your Selection: 1

Enter ID:
00017

User Found
What would you like to modify
1. Name
2. Date Joined
3. Date Terminated
4. Hourly Rate
5. Phone Number
6. Address
7. Sport Centre Code
8. Sport Center Name
9. Sport Name
Your Selection: 8

Enter Sport Centre Name
Red Bull Sport Academy
File have been updated
```

Figure 71: Modify Coach Sports Center Name

Figure 71 represent the modification to the coach's sports center name. The coach's sports center name will be replaced with the new sports center name that was entered by the administrative user. The changes will be updated to the text files immediately.

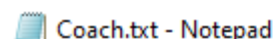


Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 450
Phone: 012-345 6789
Address: 45G, 668 Cody Ridge Road, Cleo Springs, 73729, Oklahoma
Sport Centre Code: 00010CC
Sport Center Name: Sport Pro Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0

Figure 72: Coach Sport Center Name before modification in “Coach.txt” file



Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 450
Phone: 012-345 6789
Address: 45G, 668 Cody Ridge Road, Cleo Springs, 73729, Oklahoma
Sport Centre Code: 00010CC
Sport Centre Name: Red Bull Sport Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0

Figure 73: Coach Sports Center Name after modification in “Coach.txt” file

5.1.9 Sport Name

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Coach

Please select on how you want to find the user:
1. ID
2. Fullname
Your Selection: Fullname

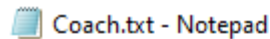
Enter Fullname:
Neke Rosberg

User Found
What would you like to modify
1. Name
2. Date Joined
3. Date Terminated
4. Hourly Rate
5. Phone Number
6. Address
7. Sport Centre Code
8. Sport Center Name
9. Sport Name
Your Selection: 9

Enter Sport Name
Formula 1
File have been updated
File have been updated
```

Figure 74: Modify Coach Sport Name

Figure 71 represent the modification to the coach sport's name. The coach sport's name will be replaced with the new coach sport's name that was entered by the administrative user. The changes will be updated to the text files immediately.



Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 450
Phone: 012-345 6789
Address: 45G, 668 Cody Ridge Road, Cleo Springs, 73729, Oklahoma
Sport Centre Code: 00010CC
Sport Centre Name: Red Bull Sport Academy
Sport Code: 00013S
Sport Name: Formula 2
Rating: 0

Figure 75: Coach's Sport Name before modification in "Coach.txt" file



Coach.txt - Notepad

File Edit Format View Help

ID: 00017
Name: Neke Rosberg
Date Joined: 25/5/2020
Date Terminated: 30/5/2021
Hourly Rate: RM 450
Phone: 012-345 6789
Address: 45G, 668 Cody Ridge Road, Cleo Springs, 73729, Oklahoma
Sport Centre Code: 00010CC
Sport Centre Name: Red Bull Sport Academy
Sport Code: 00012S
Sport Name: Formula 1
Rating: 0

Figure 76: Coach's Sport Name after modification in "Coach.txt" file

5.2 Modify Sport Record

5.2.1 Sport Name

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Sport

Please select on how you want to find the sport:
1. Sport ID
2. Sport Name
Your Selection: 1

Enter Sport ID:
00012S

User Found
What would you like to modify
1. Sport Name
2. Sport Schedule
Your Selection: Sport Name
Enter Sport Name
Formula 1
File have been updated
```

Figure 77: Modify Sport Name

Figure 77 represent the modification to the sport's name. The sport's name will be replaced with the new sport's name that was entered by the administrative user. The changes will be updated to the text files immediately.



 Sport.txt - Notepad
File Edit Format View Help
Sport ID: 00012S
Sport Name: Formular 1
Sport Schedule:
end

Figure 78: Sport Name before modification in “Sport.txt” file

 Sport.txt - Notepad
File Edit Format View Help

Sport ID: 00012S
Sport Name: Formula 1
Sport Schedule:
end

Figure 79: Sport Name after modification in “Sport.txt” file

5.2.2 Sport Schedule

```
Your Selection: Sport

Please select on how you want to find the sport:
1. Sport ID
2. Sport Name
Your Selection: 1

Enter Sport ID:
00014S

User Found
What would you like to modify
1. Sport Name
2. Sport Schedule
Your Selection: Sport Schedule

Sport Schedule:
28/5/2021 12:30 - 14:30
Enter Date you want to modify: dd/mm/yyyy
28/5/2021
Enter start time you want to modify: hh:ss (24-hours)
12:30
Enter end time you want to modify: hh:ss (24-hours)
14:30
Enter a new date: dd/mm/yyyy
25/5/2021
Enter new start time: hh:ss (24-hours)
13:00
Enter new end time: hh:ss (24-hours)
15:00
Successfully wrote to the file.
```

Figure 80: Modify Sport Schedule

Figure 80 represent the modification to the sport's schedule. The sport's schedule will be replaced with the new sport's schedule that was entered by the administrative user. The changes will be updated to the text files immediately.


 Sport.txt - Notepad
File Edit Format View Help
Sport ID: 00014S
Sport Name: Formula 3
Sport Schedule:
28/5/2021 12:30 - 14:30
end

Figure 81: Sport Schedule before modification in “Sport.txt” file


 Sport.txt - Notepad
File Edit Format View Help
Sport ID: 00014S
Sport Name: Formula 3
Sport Schedule:
25/5/2021 13:00 - 15:00
end

Figure 82: Sport Schedule after modification in “Sport.txt” file

5.3 Modify Registered Student Record

5.3.1 Name

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Student

Please select on how you want to find the user:
1. ID
2. Username
3. Fullname
Your Selection:ID

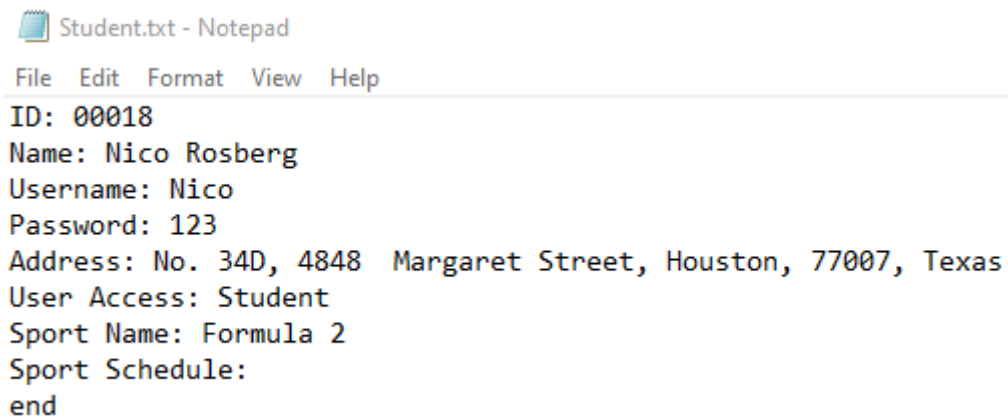
Enter ID:
00018

User Found
What would you like to modify
1. Name
2. Username
3. Password
4. Address
5. Sport Name
6. Sport Schedule
Your Selection: Name

Enter Name
Nico Hulkenberg
File have been updated
```

Figure 83: Modify Student Name

Figure 83 represent the modification to the student's name. The student's name will be replaced with the new student's name that was entered by the administrative user. The changes will be updated to the text files immediately.

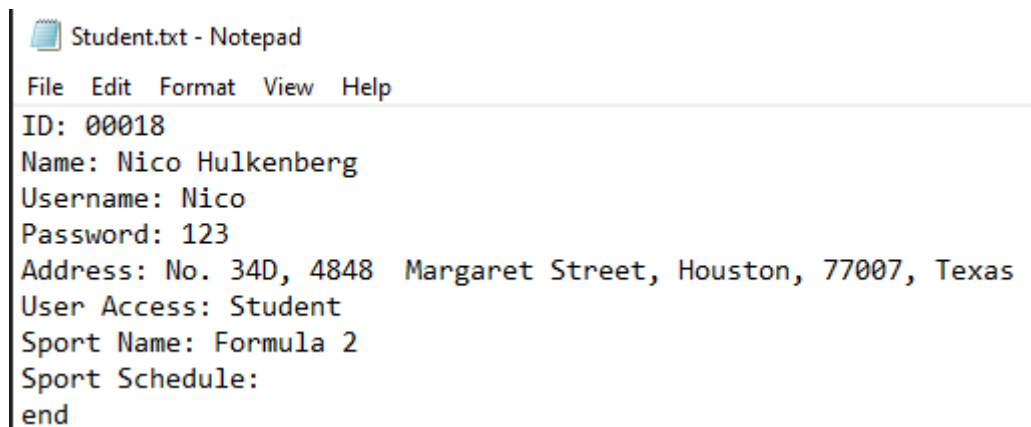


Student.txt - Notepad

File Edit Format View Help

ID: 00018
Name: Nico Rosberg
Username: Nico
Password: 123
Address: No. 34D, 4848 Margaret Street, Houston, 77007, Texas
User Access: Student
Sport Name: Formula 2
Sport Schedule:
end

Figure 84: Student Name before modification in “Student.txt” file



Student.txt - Notepad

File Edit Format View Help

ID: 00018
Name: Nico Hulkenberg
Username: Nico
Password: 123
Address: No. 34D, 4848 Margaret Street, Houston, 77007, Texas
User Access: Student
Sport Name: Formula 2
Sport Schedule:
end

Figure 85: Student Name after modification in “Student.txt” file

5.3.2 Username

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Student

Please select on how you want to find the user:
1. ID
2. Username
3. Fullname
Your Selection:2

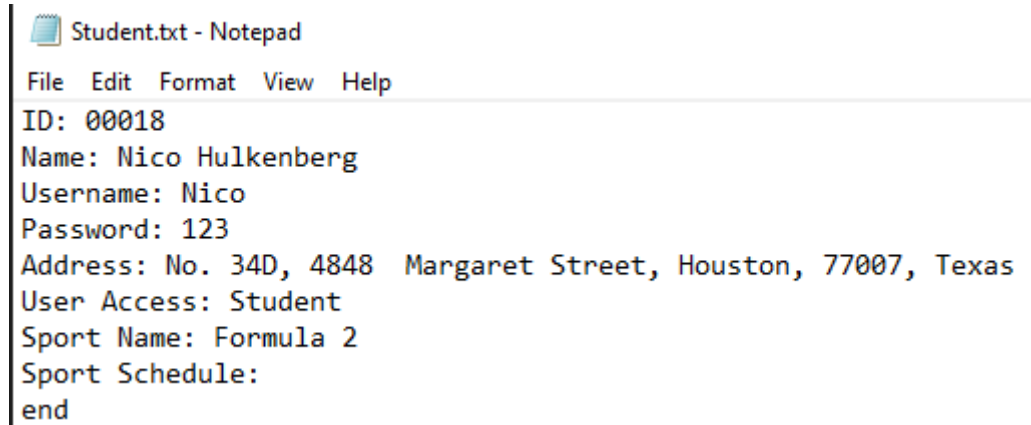
Enter Username:
Nico

User Found
What would you like to modify
1. Name
2. Username
3. Password
4. Address
5. Sport Name
6. Sport Schedule
Your Selection: Username

Enter Username
Nicholas
File have been updated
```

Figure 86: Modify Student Username

Figure 86 represent the modification to the student's username. The student's username will be replaced with the new student's username that was entered by the administrative user. The changes will be updated to the text files immediately.

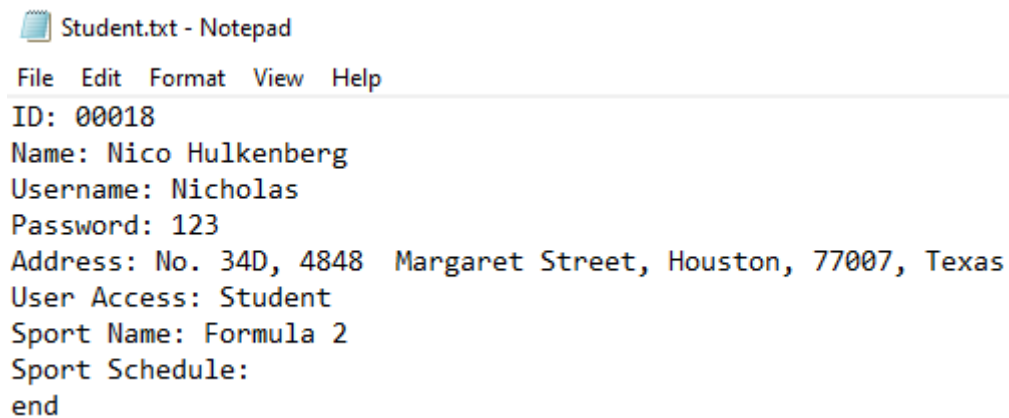


Student.txt - Notepad

File Edit Format View Help

ID: 00018
Name: Nico Hulkenberg
Username: Nico
Password: 123
Address: No. 34D, 4848 Margaret Street, Houston, 77007, Texas
User Access: Student
Sport Name: Formula 2
Sport Schedule:
end

Figure 87: Student Username before modification in “Student.txt” file



Student.txt - Notepad

File Edit Format View Help

ID: 00018
Name: Nico Hulkenberg
Username: Nicholas
Password: 123
Address: No. 34D, 4848 Margaret Street, Houston, 77007, Texas
User Access: Student
Sport Name: Formula 2
Sport Schedule:
end

Figure 88: Student Username after modification in “Student.txt” file

5.3.3 Password

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Student

Please select on how you want to find the user:
1. ID
2. Username
3. Fullname
Your Selection:3

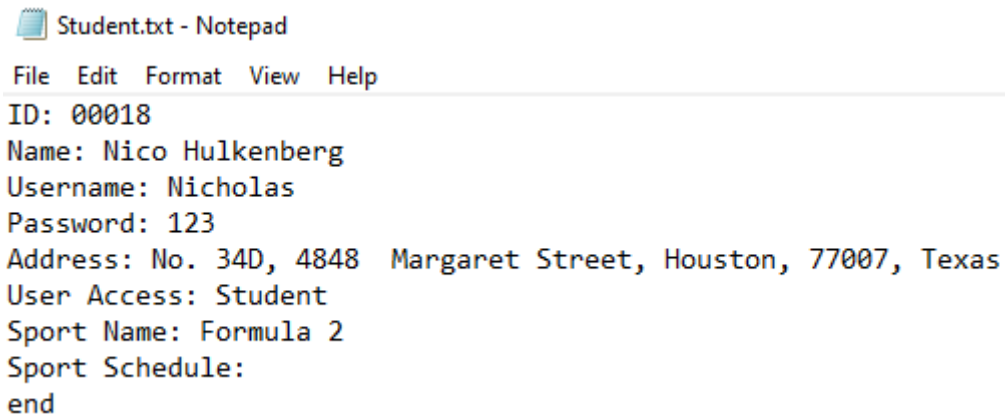
Enter Fullname:
Nico Hulkenberg

User Found
What would you like to modify
1. Name
2. Username
3. Password
4. Address
5. Sport Name
6. Sport Schedule
Your Selection: Password

Enter Password
Nick123
File have been updated
```

Figure 89: Modify Student Password

Figure 89 represent the modification to the student's password. The student's password will be replaced with the new student's password that was entered by the administrative user. The changes will be updated to the text files immediately.

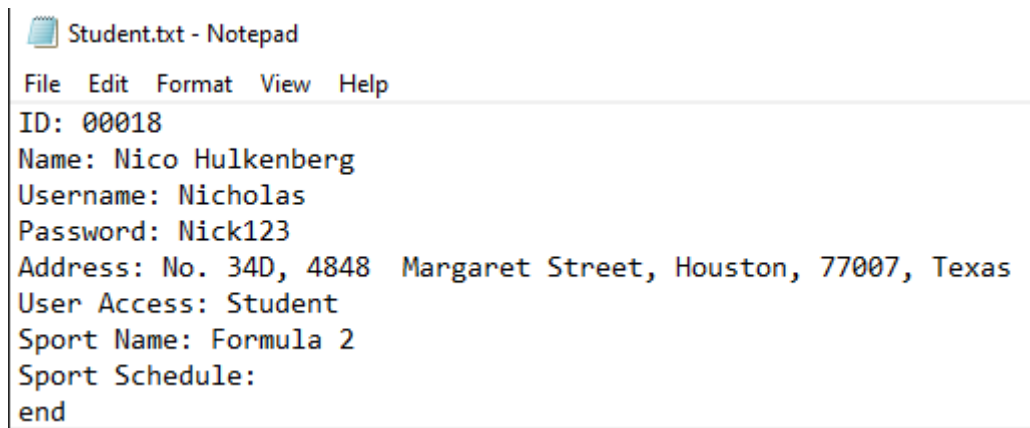


Student.txt - Notepad

File Edit Format View Help

ID: 00018
Name: Nico Hulkenberg
Username: Nicholas
Password: 123
Address: No. 34D, 4848 Margaret Street, Houston, 77007, Texas
User Access: Student
Sport Name: Formula 2
Sport Schedule:
end

Figure 90: Student Password before modification in “Student.txt” file



Student.txt - Notepad

File Edit Format View Help

ID: 00018
Name: Nico Hulkenberg
Username: Nicholas
Password: Nick123
Address: No. 34D, 4848 Margaret Street, Houston, 77007, Texas
User Access: Student
Sport Name: Formula 2
Sport Schedule:
end

Figure 91: Student Password after modification in “Student.txt” file

5.3.4 Address

```
Your Selection: Student

Please select on how you want to find the user:
1. ID
2. Username
3. Fullname
Your Selection:1

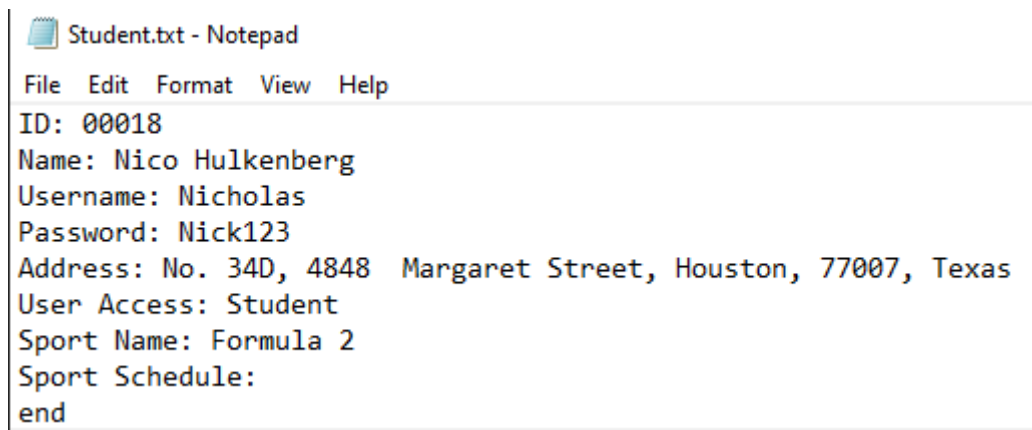
Enter ID:
00018

User Found
What would you like to modify
1. Name
2. Username
3. Password
4. Address
5. Sport Name
6. Sport Schedule
Your Selection: Address

Enter house number
No. 34F
Enter street
2700 Thrash Trail
Enter City
ROCK HALL
Enter Postal Code
21661
Enter State
Maryland
File have been updated
```

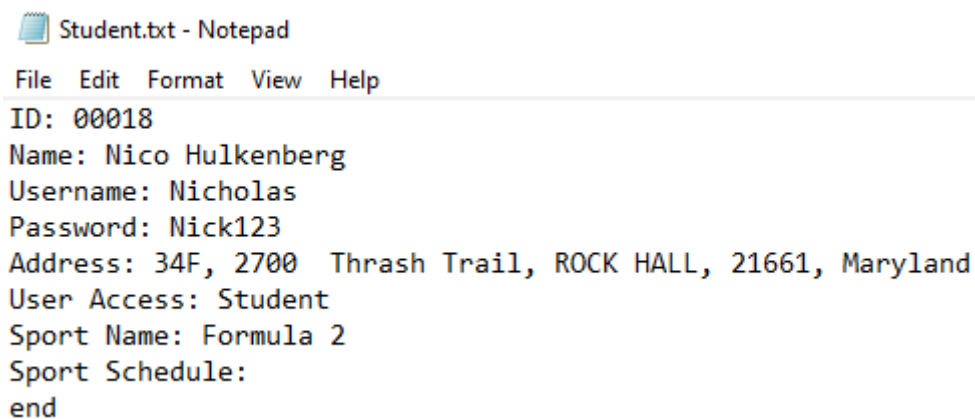
Figure 92: Modify Student Address

Figure 92 represent the modification to the student's address. The student's address will be replaced with the new student's address that was entered by the administrative user. The changes will be updated to the text files immediately.



```
Student.txt - Notepad
File Edit Format View Help
ID: 00018
Name: Nico Hulkenberg
Username: Nicholas
Password: Nick123
Address: No. 34D, 4848 Margaret Street, Houston, 77007, Texas
User Access: Student
Sport Name: Formula 2
Sport Schedule:
end
```

Figure 93: Student Address before modification in “Student.txt” file



```
Student.txt - Notepad
File Edit Format View Help
ID: 00018
Name: Nico Hulkenberg
Username: Nicholas
Password: Nick123
Address: 34F, 2700 Thrash Trail, ROCK HALL, 21661, Maryland
User Access: Student
Sport Name: Formula 2
Sport Schedule:
end
```

Figure 94: Student Address after modification in “Student.txt” file

5.3.5 Sport Name

```
Who would you like to modify
1. Coach
2. Sport
3. Registered Student
Your Selection: Student

Please select on how you want to find the user:
1. ID
2. Username
3. Fullname
Your Selection:1

Enter ID:
00018

User Found
What would you like to modify
1. Name
2. Username
3. Password
4. Address
5. Sport Name
6. Sport Schedule
Your Selection: 5

Enter Sport Name
Formula 1
File have been updated
```

Figure 95: Modify Student Sport Name

Figure 95 represent the modification to the student's sport name. The student's sport name will be replaced with the new student's sport name that was entered by the administrative user. The changes will be updated to the text files immediately.


 Student.txt - Notepad
File Edit Format View Help
ID: 00018
Name: Nico Hulkenberg
Username: Nicholas
Password: Nick123
Address: 34F, 2700 Thrash Trail, ROCK HALL, 21661, Maryland
User Access: Student
Sport Name: Formula 2
Sport Schedule:
end

Figure 96: Student Sport Name before modification in “Student.txt” file


 Student.txt - Notepad
File Edit Format View Help
ID: 00018
Name: Nico Hulkenberg
Username: Nicholas
Password: Nick123
Address: 34F, 2700 Thrash Trail, ROCK HALL, 21661, Maryland
User Access: Student
Sport Name: Formula 1
Sport Schedule:
end

Figure 97: Student Sport Name after modification in “Student.txt” file

5.3.6 Sport Schedule

```
Please select on how you want to find the user:
1. ID
2. Username
3. Fullname
Your Selection:1

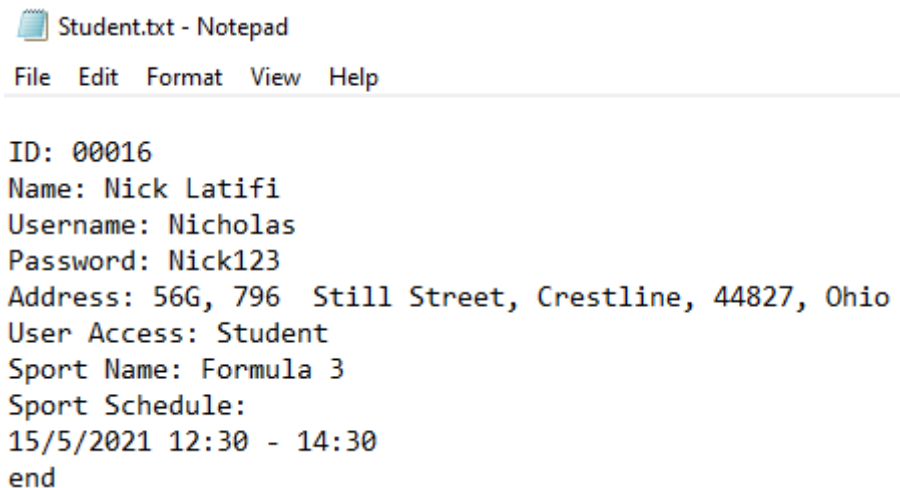
Enter ID:
00016

User Found
What would you like to modify
1. Name
2. Username
3. Password
4. Address
5. Sport Name
6. Sport Schedule
Your Selection: 6

Sport Schedule:
15/5/2021 12:30 - 14:30
Enter Date you want to modify: dd/mm/yyyy
15/5/2021
Enter start time you want to modify: hh:ss (24-hours)
12:30
Enter end time you want to modify: hh:ss (24-hours)
14:30
Enter a new date: dd/mm/yyyy
25/5/2021
Enter new start time: hh:ss (24-hours)
13:00
Enter new end time: hh:ss (24-hours)
15:00
Successfully wrote to the file.
```

Figure 98: Modify Student Sport Schedule

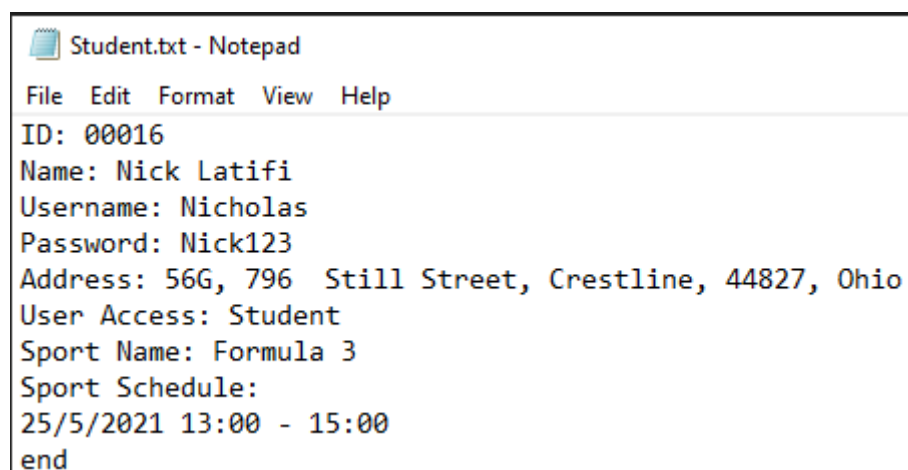
Figure 98 represent the modification to the student's sports schedule. The student's sports schedule will be replaced with the new student's sports schedule that was entered by the administrative user. The changes will be updated to the text files immediately.



```
Student.txt - Notepad
File Edit Format View Help

ID: 00016
Name: Nick Latifi
Username: Nicholas
Password: Nick123
Address: 56G, 796 Still Street, Crestline, 44827, Ohio
User Access: Student
Sport Name: Formula 3
Sport Schedule:
15/5/2021 12:30 - 14:30
end
```

Figure 99: Student Sports Schedule before modification in “Student.txt” file



```
Student.txt - Notepad
File Edit Format View Help

ID: 00016
Name: Nick Latifi
Username: Nicholas
Password: Nick123
Address: 56G, 796 Still Street, Crestline, 44827, Ohio
User Access: Student
Sport Name: Formula 3
Sport Schedule:
25/5/2021 13:00 - 15:00
end
```

Figure 100: Student Sport Schedule after modification in “Student.txt” file

6. Exit Function

```
Welcome, Loke Weng Khay

What would you like to do today
1. Add Record
2. Display All Records
3. Search Specific Record
4. Sort Record
5. Modify Record
6. Exit
Your Selection: 6
BUILD SUCCESSFUL (total time: 6 minutes 4 seconds)
```

Figure 101: Exit Function

Figure 101 represent the exit function in the Administrative Account.

Registered Student Account

```
You selection: 1
You selected: Login

Enter username
Max
Enter password
123

Welcome, Max Verstappen

What would you like to do today?
1. View Coach
2. View Self-Record
3. View Sport Schedule
4. Add Sport Schedule
5. Rate and Comment Coach
6. Exit
Your Selection: |
```

Figure 102: Registered Student Account in the Real Champion Sport Academy System

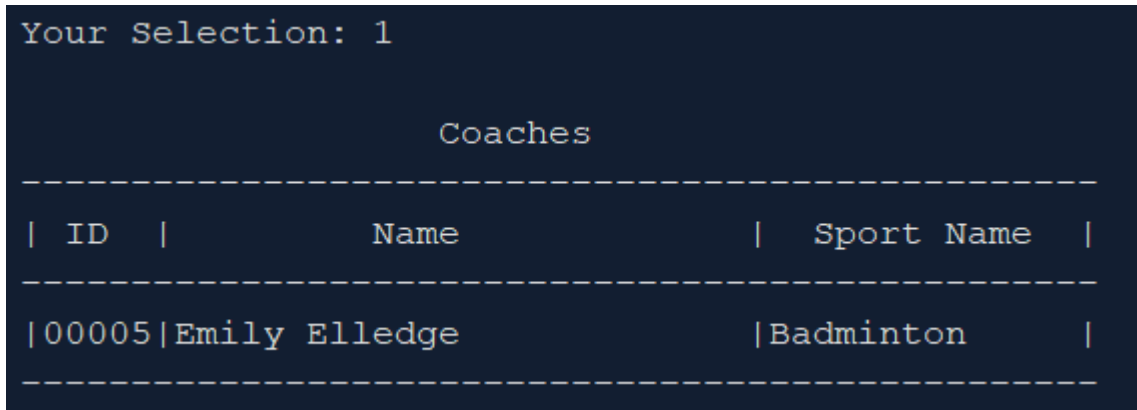
Figure 102 represent the student account, where they can manage their own account.

Some of the function available in the Registered Student account is:

- View Coach
- View Self-Record
- View Sport Schedule
- Add Sport Schedule
- Rate and Comment Coach
- Exit

Registered Student Account Functions:

1. View Coaches Function



```

Your Selection: 1

Coaches
-----
| ID      | Name                | Sport Name |
-----
| 00005   | Emily Elledge       | Badminton  |
-----

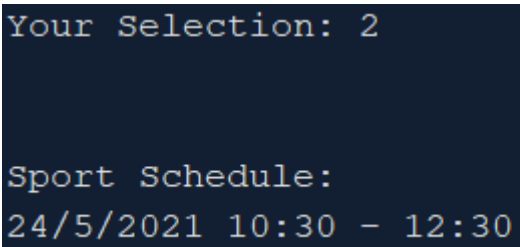
```

ID	Name	Sport Name
00005	Emily Elledge	Badminton

Figure 103: View Coach Based on Student's Sport

In the View Coach function, the student account can view their sport's coach names based on the student selected sports.

2. View Self-Records Function



```
Your Selection: 2  
  
Sport Schedule:  
24/5/2021 10:30 - 12:30
```

Figure 104: View Self-Records Function

In the View Self-Record function, the student account can view their own timetable or sport schedule that the student registered.

3. View Sport Schedule Function

```
Your Selection: 3

Sport Name: Badminton
Sport Schedule:
15/5/2021 19:00 - 23:00
24/5/2021 10:30 - 12:30
```

Figure 105: View All Sport Schedule Based on the Student Sport

In the View Sport Schedule Function, the student account can view all available class timetable based on the student selected sport.

4. Add Sport Schedule Function

```
Your Selection: 4

Avaliable time for:
Sport Name: Badminton
Sport Schedule:
15/5/2021 19:00 - 23:00
24/5/2021 10:30 - 12:30

Enter Date for Sport Name: Badminton: dd/mm/yyyy
15/5/2021
Enter start time for the class: hh:ss (24-hours)
19:00
Enter end time for the class: hh:ss (24-hours)
23:00
Added Schedule: 15/5/2021 19:00 - 23:00
15/5/2021 19:00 - 23:00
Successfully wrote to the file.
```

Figure 106: Add Sport Schedule Based on the Student's Sport

The student can also register and add available class to their timetable. Once added, the record will be saved into the "Student.txt" file.



```
Student.txt - Notepad
File Edit Format View Help
ID: 00011
Name: Max Verstappen
Username: Max
Password: 123
Address: No. 2154, Desert Broom Court, GRAND RAPIDS, 49530, Michigan
User Access: Student
Sport Name: Badminton
Sport Schedule:
24/5/2021 10:30 - 12:30
15/5/2021 19:00 - 23:00
end
```

Figure 107: "Student.txt" file after code execution in figure 106

5. Rate and Comment Coach Function

```
Your Selection: 5

                                Coach Rating
-----
| ID |           Name           | Sport Name |
-----
|00005|Emily Elledge         |Badminton  |
-----

Enter coach full name to rate:
emily elledge
Please enter rating: (1-5)
2
Any Comments
She is bad!
Successfully wrote to the file.
File have been updated
```

Figure 108: Rate and Comment Coach Function

The student can also rate their own coach, and the rating will be anonymous. The rating data will be store in the “Rating.txt” file, and the Coach rating also will be updated in the “Coach.txt” file immediately.



Rating.txt - Notepad

File Edit Format View Help

ID: 00015

Name: Sergio

Rating: 5

Comments: Awesome

ID: 00005

Name: Emily Elledge

Rating: 4

Comments: Ok

|

ID: 00005

Name: Emily Elledge

Rating: 2

Comments: She is bad!

Figure 109: Coach Rating and comments in “Rating.txt” file

ID: 00005

Name: Emily Elledge

Date Joined: 20/11/2020

Date Terminated: -

Hourly Rate: RM 150

Phone: 012-482 6754

Address: No.67 , Unrelated Roads

Sport Centre Code: 00021CC

Sport Center Name: Sport Pros

Sport Code: 00002S

Sport Name: Badminton

Rating: 3

Figure 110: Updated Rating value in the “Coach.txt” file

6. Exit Function

```
What would you like to do today?  
1. View Coach  
2. View Self-Record  
3. View Sport Schedule  
4. Add Sport Schedule  
5. Rate and Comment Coach  
6. Exit  
Your Selection: 6  
  
BUILD SUCCESSFUL (total time: 8 minutes 34 seconds)  
|
```

Figure 111: Exit Function

Figure 111 represents the student account's exit function to allow the student to exit the system.

2. Sign up Function

```
You selection: 2
You selected: Sign Up

Student Information:
Enter Full Name
Judith M Carlson
Enter Username:
Judith
Enter Password:
Jo456
Re-Enter Password again
Jo456
Enter Sport Name
Cricket
Enter house number
No. 12H
Enter street
2641 Green Acres Road
Enter City
Greenville
Enter Postal Code
27834
Enter State
North Carolina
```

Figure 112: Sign Up function

The sign-up feature will be available on the front page of the system, where a newcomer can register to the system.



```
Student.txt - Notepad
File Edit Format View Help
ID: 00019
Name: Judith M Carlson
Username: Judith
Password: Jo456
Address: No. 12H, 2641 Green Acres Road, Greenville, 27834, North Carolina
User Access: Student
Sport Name: Cricket
Sport Schedule:
end
```

Figure 113: New Student data added to “Student.txt” file after execution of figure 112

Once the sign-up process is done, the new student data will be saved into the Real Champions Sport Academy System, and they can log in to the system.

3. View Sport Function

```
You selection: 3
You selected: View Sport

All available sports
-----
| Sport Name |
-----
|Swimming    |
-----
|Badminton   |
-----
|Football    |
-----
|Archery     |
-----
|Gymnastics  |
-----
|Volleyball  |
-----
|Basketball  |
-----
|Cricket     |
-----
|Tennis      |
-----
|Table Tennis|
-----
|Bowling     |
-----
|Formula 1   |
-----
|Formula 2   |
-----
|Formula 3   |
-----
```

Figure 114: View All available sports in the Real Champions Sport Academy System

In figure 114, non-registered user can view all available sport in Real Champions Sport Academy System.

4. View Sport Schedule Function

```
You selection: 4
You selected: View Sport Schedule

Enter the sport you would want to view the sport schedule
Formula 2

Sport Name: Formula 2
Sport Schedule:
23/5/2021 12:30 - 14:30
29/5/2021 14:30 - 16:30
```

Figure 115: View sport schedule based on the selected sport

In figure 115, non-registered user can also view sports schedule based on the selected sport in Real Champions Sport Academy System.

5. Exit Function

```
You selection: 5  
You selected: Exit  
  
BUILD SUCCESSFUL (total time: 3 seconds)
```

Figure 116: Exit Function

Figure 116 is the exit function for the user on the front page of the Real Champions Sport Academy System.

6. Conclusion

The Real Champions Sport Academy System was a challenge as it was a test of all my skills that I have learned and acquired in my Object-Oriented Programming in Java classes. This program has taught me the importance of understanding object-oriented programming during the development of this software. Java programming has been an essential skill to obtain, as most Android apps run in java code. Learning java with object-oriented programming will help me in the future for future projects in other programming languages.

For this assignment, I have met most of the requirement of this project, but I felt that I could have done this project better. But during the development of this project, I have encountered several challenges.

One of the challenges is a lack of time. This is because the project size is enormous and has to be completed in 2 and a half months. This is not the only challenge I faced as this is not the only assignment that I have to complete in time, as I have other course assignments to be completed in the time given.

With the help of my time management skills, I managed my time between multiple assignment projects and hand in the work or assignment in a timely manner.

References

Anon., 2021. *GeeksForGeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/inheritance-in-java/>
[Accessed 7 May 2021].

Anon., n.d. *Guru 99*. [Online]

Available at: <https://www.guru99.com/java-class-inheritance.html>
[Accessed 7 May 2021].

Anon., n.d. *Guru 99*. [Online]

Available at: <https://www.guru99.com/java-oops-concept.html>
[Accessed 7 May 2021].

Elliott, E., 2018. *JavaScript Scene*. [Online]

Available at: [https://medium.com/javascript-scene/the-forgotten-history-of-oop-88d71b9b2d9f#:~:text=%E2%80%9CObject%2DOriented%20Programming%E2%80%9D%20\(his%20Sketchpad%20Thesis%20in%201963.](https://medium.com/javascript-scene/the-forgotten-history-of-oop-88d71b9b2d9f#:~:text=%E2%80%9CObject%2DOriented%20Programming%E2%80%9D%20(his%20Sketchpad%20Thesis%20in%201963.)
[Accessed 7 May 2021].

Kölling, M., 1999. *Kent Academic Repository*. [Online]

Available at: https://kar.kent.ac.uk/21879/2/the_problem_of_teaching_object-oriented_kolling_1.pdf
[Accessed 7 May 2021].

Leahy, P., 2019. *ThoughtCo.* [Online]

Available at: <https://www.thoughtco.com/aggregation-2033995#:~:text=Aggregation%20in%20Java%20is%20a,%22whole%2Fpart%22%20relationship.&text=Each%20class%20referenced%20is%20considered,references%20in%20an%20aggregation%20relationship.>
[Accessed 7 May 2021].

Nishadha, 2020. *creately*. [Online]

Available at: <https://creately.com/blog/diagrams/class-diagram-relationships/>
[Accessed 7 May 2021].