NETBALL TEAM SELECTOR
OWAIN LANSDOWNE
CANDIDATE NUMBER-9503
CENTRE – GODALMING COLLEGE
CENTRE NUMBER - 64395

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

NETBALL TEAM SELECTOR

CONTENTS

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

Figure Table

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

RESEARCH

Rodborough Rockets is a Surrey based Netball club for ladies which plays on a Tuesday evening. They are not part of a league but instead play a non-competitive game between themselves. Currently they just split the ladies that turn up into teams and have an hour-long game where they rotate positions every so often.

The task is to allow one player to enter into a program who is attending and then split the players who can attend into two approximately even teams and assign each player a position. If less than 14 players can attend, then the relevant number of positions will be removed from the formation and the remaining positions allocated accordingly. However, if more than 14 players can turn up then some of the players need to be a substitute for some of the time. The other part of the task is to inform the user how much each player owes to play each week as it varies between players. This differs according to some having England Netball Membership and how many weeks they have played in the year so far. Once who can attend has been recorded it needs to display how much each needs to pay.

Currently one player roughly splits the players up into teams at the start of the session instead of taking into account the players' abilities which can lead to some unfair teams. They record who has attended each week on a piece of a paper and so they use that to know how much each should be paying but if they were to lose the paper then they wouldn't know who owes what.

Who – Rodborough Rockets
What – Netball Team
Main End User – Jenny Lansdowne (Rodborough Rockets Secretary)
Additional End Users - Chairperson and treasurer

In summary the project involves splitting the players into two evenly matched teams so that they can have close fun games. The other part of the project is recording which players have played each week and whether they have Netball England Membership so that everyone knows how much they each owe. I am tackling this problem so that they can have close even matches as well as making it easier for everyone to know how much they owe.

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

INTERVIEW

I have conducted an Interview with the Rodborough Rockets Secretary and Player Jenny Lansdowne to understand what their current system is and how it works and to see how they wish to see the program work.

What is the current system for splitting the players into two teams?

Two people are given a set of bibs each and they hand them out to people.

Do you split the players in accordance of ability to make the teams even?

The people giving out the bibs normally do try to get a good mix (rather than giving their bibs to all the good players). We would not want to rank people according to ability though. The kinder way is to try to get a mix of heights on each team.

Is having even teams more important than mixing the teams up each week?

Yes, we are playing for fun so everyone wants to have an enjoyable game. No-one wants a one-sided game where one team is getting thrashed.

Do you record what the teams were all the previous weeks?

No, there is no record made from previous weeks.

Normally how many players turn up each week?
This is the problem – it can vary so much. Anything from 6 to 20.

How many players are normally on a netball team?
7 a side

How many players are part of the club?
Difficult to say as it is pay and play, so we have people on our books who have not been for ages, but may suddenly turn up. Also, new people are welcome. At the moment we probably have about 30 players on our register, but not all come regularly and we never get everyone together.

What are the positions?
Goal shooter (GS), Goal Attack (GA), Wing Attack (WA), Centre (C), Wing Defence (WD), Goal Defence (GD), Goal Keeper (GK)

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

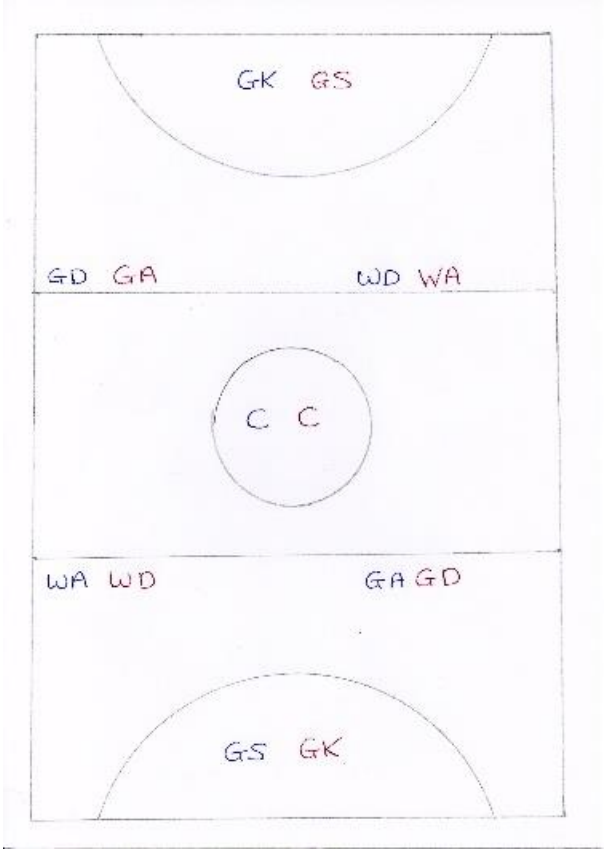


Figure 1 - Netball court with the positions.

What happens if there are less than 14 players?

We have to start dropping positions. The way it works is that if we have an odd number of players (less than 14), we have one floating Centre, who either team can throw to in attack, but who will never defend. So:

13 = floating C

12 = drop WD x2

11= floating C, drop WD x2

10 = drop WD x2 & WA x2

9 = floating C, drop WD x2 & WA x2

8 = half court game, bibs are GS x2, GA x2, WA x2, C x2 (team starts either in attack or defence, if defending team intercepts must pass ball back past half way before can turn round and attack)

7 = half court as above, but with floating C

6 = half court as 8, but drop WA x2

Less than 6 players would really be difficult.

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

What happens if there are more than 14 players?

People have to start sitting out for one position. So if 15 or 16, one or two people would sit out after playing centre. If 17 or 18, one or two people would also sit out after playing GK. If 19 and above, it becomes more awkward (luckily very rare).

What amount of players would enable you to have 4 teams?

If we have 20 or more, we could have 4 teams of 5. This works when we play outside as we have access to two courts. Inside we only have one court so would try to avoid 4 teams.

If we had 21, we could do three teams of 7 inside.

How long do you play for?

We have an hour session. Allowing for a couple of minutes changeover between positions, we usually play 7 minutes per position.

Do you need each player to enter whether they can attend or just one person to enter it for everyone?

Just one person taking the register, usually when the subs are being collected.

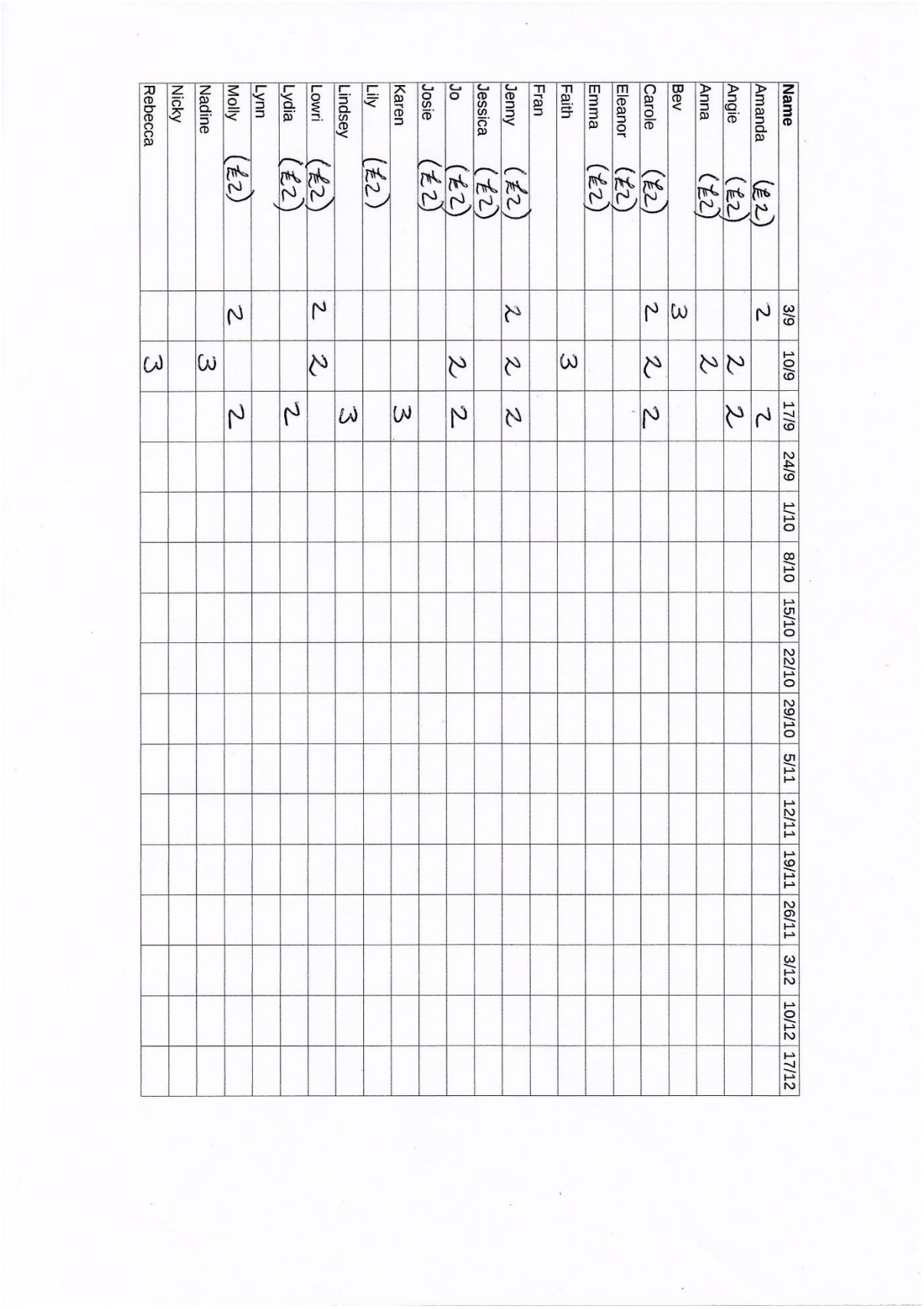
We have a slightly different system for subs now, because people who have taken out England Netball membership are entitled to a £21 refund from the club. This is done by those players paying £2 subs rather than £3 subs for the first 21 weeks that they play in the year. Is it possible for this to be calculated by the system and to know when they need to pay full price again?

What is your current system for recording subs paid and the amount required to pay?

It is done on a piece of paper like a register. When someone attends and pays their subs, we write in the column for that date whether they have paid £3 (no insurance), £2 (with insurance) or £1 (children). For those people paying £2, we have a second piece of paper where we write the date that they attended. This has 21 columns, so when they get to the end they go back to paying £3.

Example of how the subs paid by each person are recorded. (Names in the records have been changed to comply with data protection rules.)

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395



Name	3/9	10/9	17/9	24/9	1/10	8/10	15/10	22/10	29/10	5/11	12/11	19/11	26/11	3/12	10/12	17/12
Amanda (£2)	2		2													
Angie (£2)		2	2													
Anna (£2)		2														
Bev	3															
Carole (£2)	2	2	2													
Eleanor (£2)																
Emma (£2)																
Faith		3														
Fran																
Jenny (£2)	2	2	2													
Jessica (£2)																
Jo (£2)		2	2													
Josie (£2)																
Karen			3													
Lily (£2)																
Lindsey			3													
Lowri (£2)	2	2														
Lydia (£2)			2													
Lynn																
Molly (£2)	2		2													
Nadine		3														
Nicky																
Rebecca	3															

Figure 2 - Current method of recording who turns up

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

Sally		3														
Sam			3													
Stella (£2)	2															
Vanessa																
Young People																
Gabi	1															
Georgina			1													
Isabelle																
Natasha	1															

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

Figure 3 - Currently how they record who turns up continued

Insurance refund	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Amanda	3/9	17/9																			
Angie	10/9	17/9																			
Anna	10/9																				
Carole	3/9	10/9	17/9																		
Eleanor																					
Emma																					
Jenny	3/9	10/9	17/9																		
Jessica																					
Jo	10/9	17/9																			
Josie																					
Lily																					
Lowri	3/9	10/9																			
Lydia	17/9																				
Molly	3/9	17/9																			
Stella	3/9																				

Figure 4 - Chart to record when those with insurance have paid £2 for 21 weeks because then they will have to pay £3

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

SURVEY WITH THE PLAYERS

Would it be helpful if the teams were to be generated for you?

Sees whether it would helpful for the teams to be generated beforehand.

- Yes
- No

How do you want the teams to be generated?

Finds out what the players want the teams to be sorted and generated by.

- Based on positions
- Based on height
- Done randomly

Are you happy to enter online whether you can turn up?

Checks whether the players are happy to enter whether they can turn up.

- Yes
- No

Do you want to be told how much you need to pay that week?

Sees whether the players want to be told how much they owe.

- Yes
- Don't mind

SURVEY RESULTS

Would it be helpful if the teams were to be generated for you?	
Yes	14
No	2
How do you want the teams to be generated?	
Based on positions	8
Based on height	5
Done randomly	3
Are you happy to enter online whether you can turn up?	
Yes	11
No	5
Do you want to be told how much you need to pay that week?	
Yes	13
Don't mind	3

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

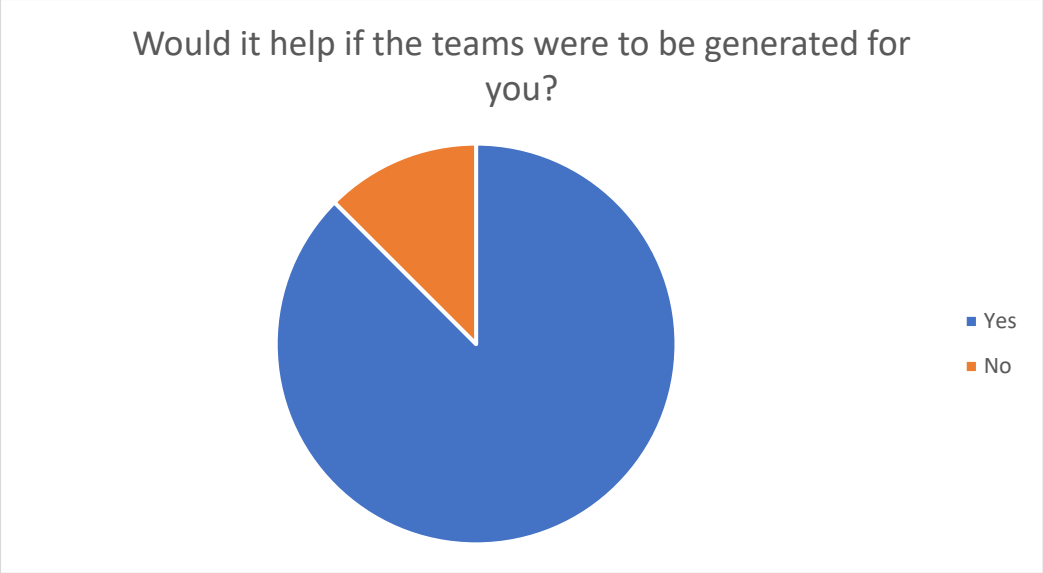


Figure 5 - Pie chart showing whether the players want the teams to generated

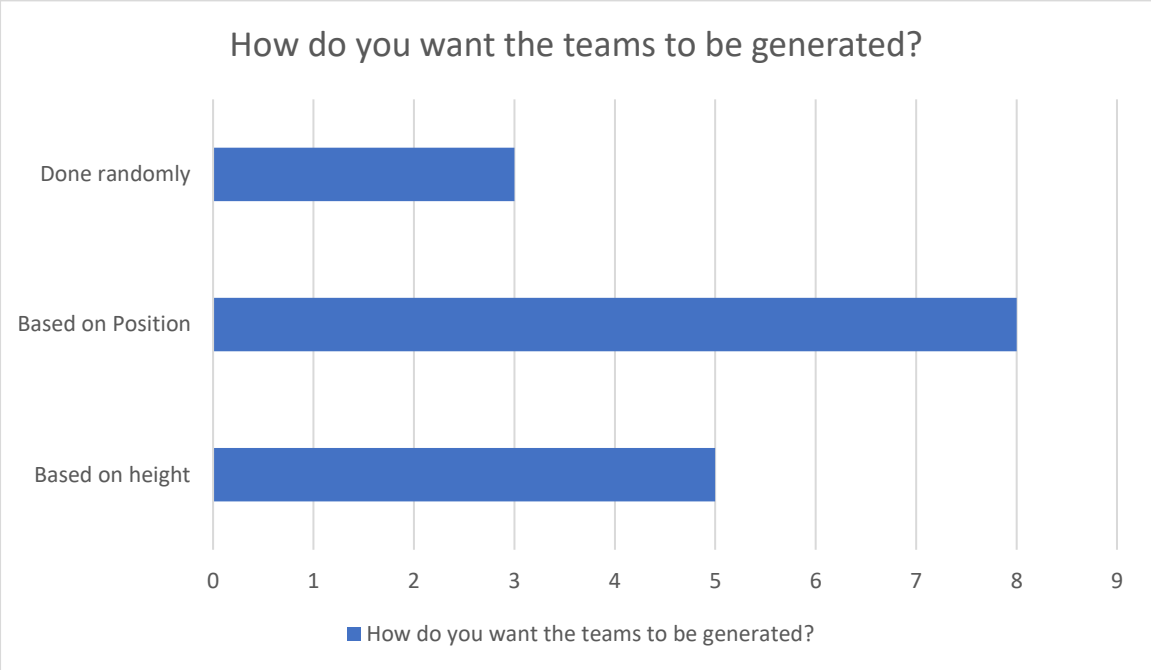


Figure 6 - Bar Chart showing players preferences on how they want teams to be generated

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

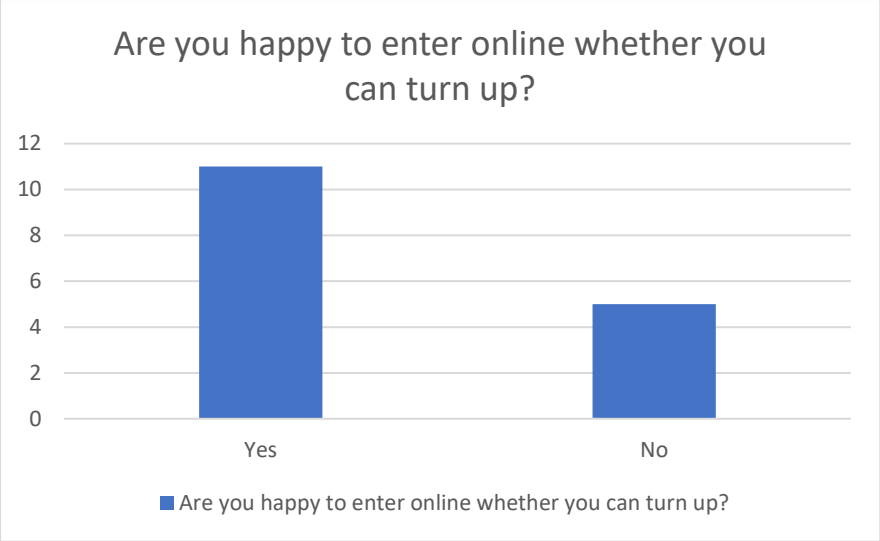


Figure 7 - Bar chart showing how many players are happy to enter their availability online

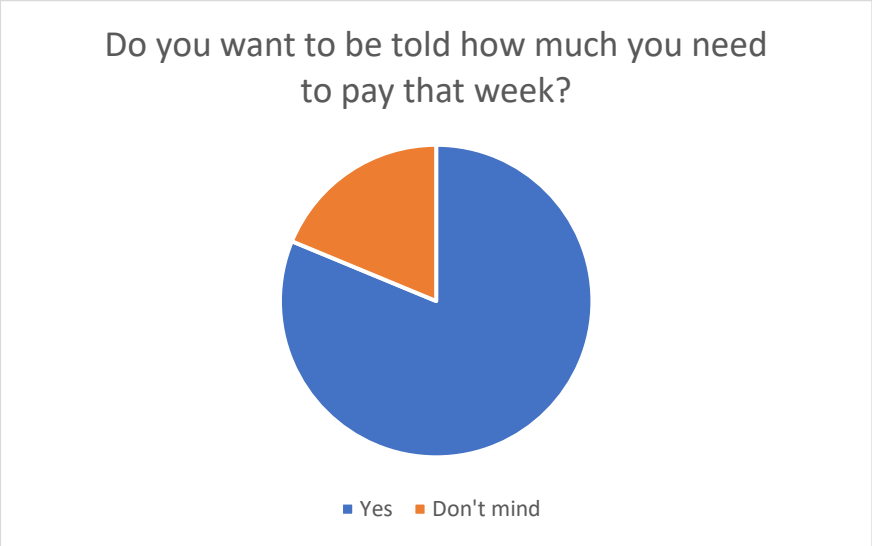


Figure 8 - Pie Chart showing whether players want to be told how much they need to pay

I conclude that the players would like the teams to be generated beforehand and the teams to be generated in accordance to the players preferred positions as that way they can have the most fun by playing one of their favourite position as well as having closer games if people are playing their preferred position.

ANALYSIS

LIST OF USERS

This is the list of users that will be using the program.

Users		**Role**
Organisation	**Person**	
Rodborough Rockets	Manager	Generate the teams when people have submitted whether they can attend and edit players' details if they get insurance.
Players	Each of the players (About 20 players)	Confirm whether they can attend this week and view how much they owe this week to play.

IPSO DIAGRAM

An IPSO diagram to show who will be using the program, what processes there will be in the program, what data needs to be stored and then the data which will be outputted.

Input	**Process**	**Storage**	**Output**
Player's availability for that week	Add Player	Database tables	Each team
Press button to generate teams	Modifies Player's Details	Player's details	Price owed by each player
	Remove Player	Player's Insurance details	
	Add Session	Player of the session	
	Add Attendance	Session	
	Calculates amount due if the player is attending the session	Player's Preferred Positions	
	Generates Team	Position	
	Awards Player of the Session		
	Sends out information about new session in an email		
	Sends out information about generated team in an email		

CASE DIAGRAM

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

A case diagram to show the processes and which process will be used by each group of users.

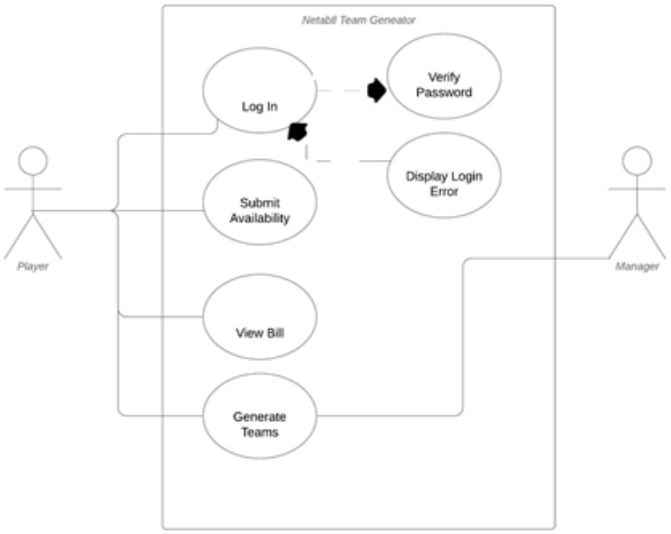


Figure 9 - Case Diagram

FLOWCHART

PAYMENT CALCULATOR

A flowchart to show how they work out the amount owed by each player according to whether they are a child or an adult as well as whether they have insurance.

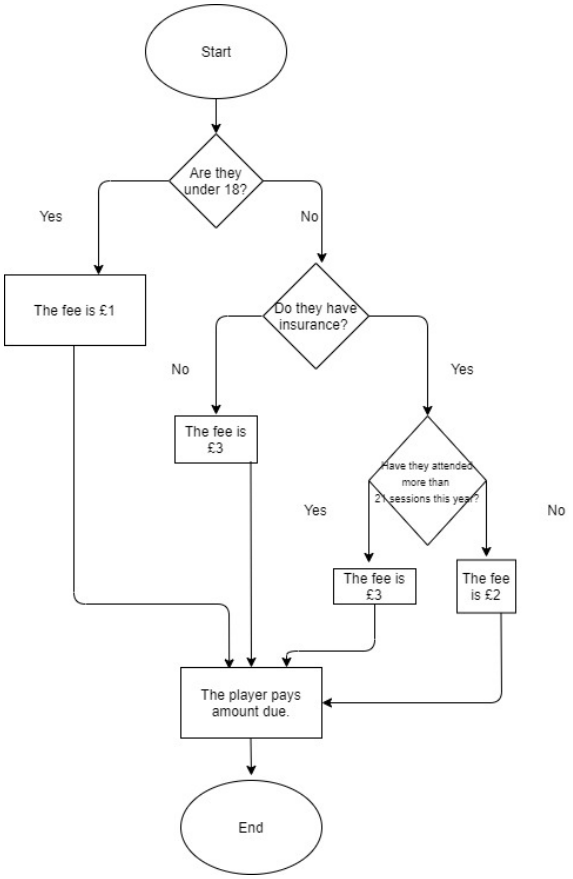
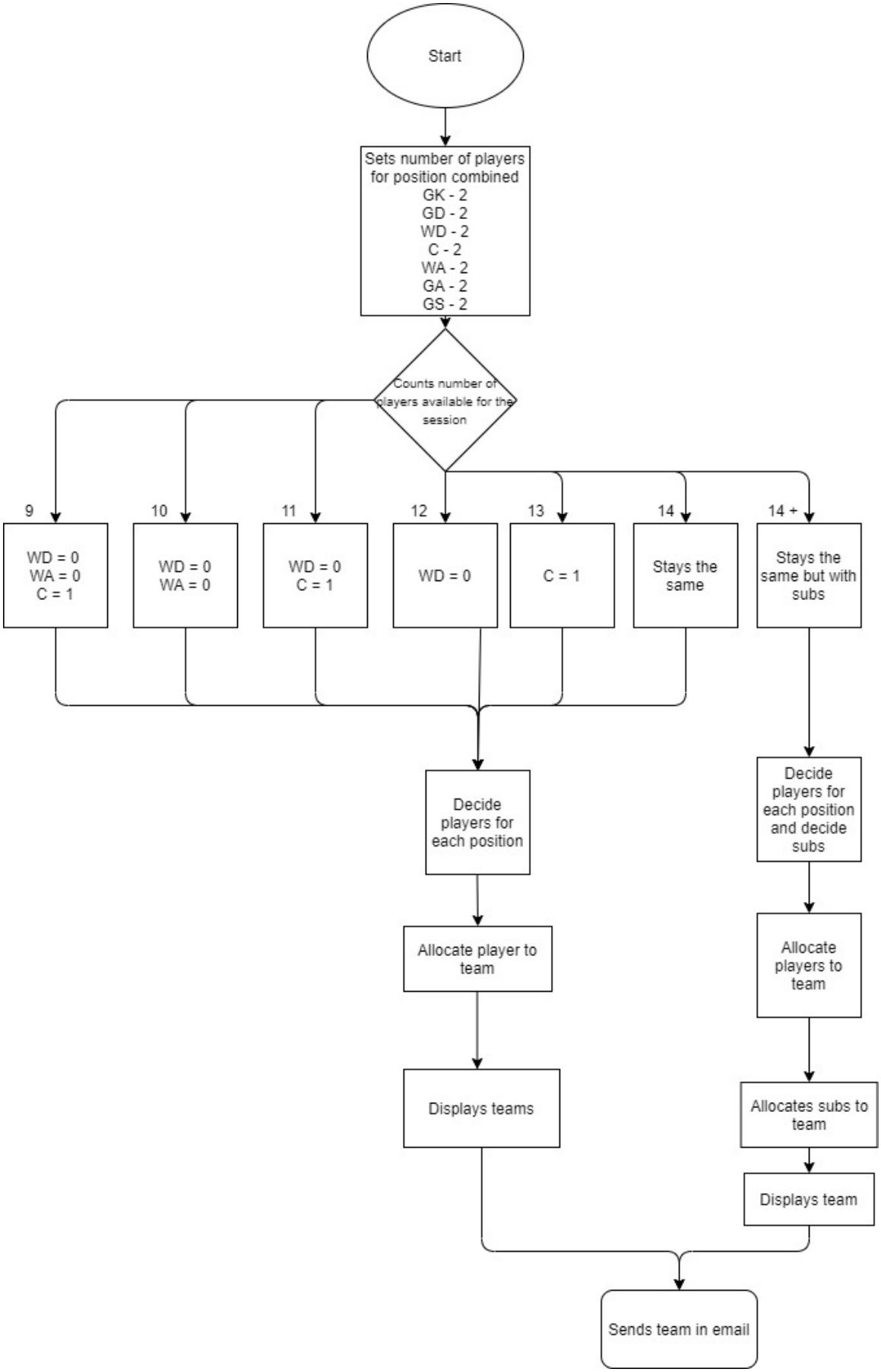


Figure 10 - Payment Calculator flowchart

TEAM SELECTOR FLOWCHART

How they split the players into two teams depending on how many players are playing and so which positions are dropped.

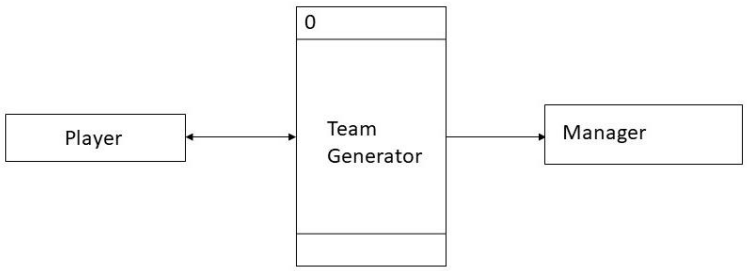


Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

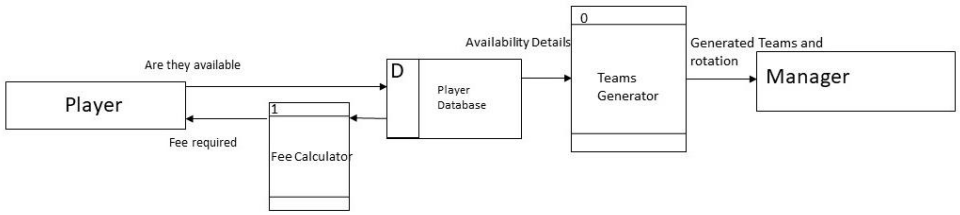
Figure 11 - Team generator flowchart

UML DATA-FLOW DIAGRAMS

This data flow diagram shows how the data flows in the system between the players and the manager.



This shows the data flow in more detail showing more processes and showing what the data is that is going between the processes and the players and manager.



DOCUMENT SPECIFICATION SHEET (EXISTING REGISTER FOR ADULTS)

Volumetrics				
Document description	System	Document	Name	Sheet
Register	Netball Adult Register	1	Owain Lansdowne	1
Stationery ref.	Size	Number of parts	Method of preparation	
Register	A4 Landscape	1	Filled in by hand	
Filing sequence	Medium	Prepared by		
Not filed	Paper	Club Manager		
Frequency of preparation	Retention period	Location of file		
Once per term	Not retained	Manager's house		
> o	Minimum	Maximum	Av/Abs	Growth rate/fluctuations

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

	0	1 per session	1 per session	

Users/receipts	Purpose	Frequency of use
Netball Managers	Records who attends and how much they paid.	Once a session, with a session every week.

Data Dictionary

Ref	Name	Data Type	Length	Occurrence	Source of data
1	Name	String	10	Per Entry	Netball Secretary
2	Amount Due	Integer	1	Per Entry	Netball Secretary
3	3/9	Integer	1	Per Entry	Netball Manager
4	10/9	Integer	1	Per Entry	Netball Manager
5	17/9	Integer	1	Per Entry	Netball Manager
6	24/9	Integer	1	Per Entry	Netball Manager
7	1/10	Integer	1	Per Entry	Netball Manager
8	8/10	Integer	1	Per Entry	Netball Manager
9	15/10	Integer	1	Per Entry	Netball Manager
10	22/10	Integer	1	Per Entry	Netball Manager
11	29/10	Integer	1	Per Entry	Netball Manager
12	5/11	Integer	1	Per Entry	Netball Manager
13	12/11	Integer	1	Per Entry	Netball Manager
14	19/11	Integer	1	Per Entry	Netball Manager
15	26/11	Integer	1	Per Entry	Netball Manager
16	3/12	Integer	1	Per Entry	Netball Manager
17	10/12	Integer	1	Per Entry	Netball Manager
18	17/12	Integer	1	Per Entry	Netball Manager

DOCUMENT SPECIFICATION SHEET (EXISTING REGISTER FOR UNDER 16S)

Volumetrics

Document description	System	Document	Name	Sheet
Register	Netball Youth Register	1	Owain Lansdowne	1

Stationery ref.	Size	Number of parts	Method of preparation
Register	A4 Landscape	1	Filled in by hand

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

Filing sequence	Medium	Prepared by
Not filed	Paper	Club Manager
Frequency of preparation	**Retention period**	**Location of file**
Once per term	Not retained	Manager's house

Volume	Minimum	Maximum	Av/Abs	Growth rate/fluctuations
	0	1 per session	1 per session	

Users/receipts	Purpose	Frequency of use
Netball Managers	Records who attends and how much they paid.	Once a session, with a session every week.

Data Dictionary

Ref	Name	Data Type	Length	Occurrence	Source of data
1	Name	String	10	Per Entry	Netball Secretary
2	3/9	Integer	1	Per Entry	Netball Manager
3	10/9	Integer	1	Per Entry	Netball Manager
4	17/9	Integer	1	Per Entry	Netball Manager
5	24/9	Integer	1	Per Entry	Netball Manager
6	1/10	Integer	1	Per Entry	Netball Manager
7	8/10	Integer	1	Per Entry	Netball Manager
8	15/10	Integer	1	Per Entry	Netball Manager
9	22/10	Integer	1	Per Entry	Netball Manager
10	29/10	Integer	1	Per Entry	Netball Manager
11	5/11	Integer	1	Per Entry	Netball Manager
12	12/11	Integer	1	Per Entry	Netball Manager
13	19/11	Integer	1	Per Entry	Netball Manager
14	26/11	Integer	1	Per Entry	Netball Manager
15	3/12	Integer	1	Per Entry	Netball Manager
16	10/12	Integer	1	Per Entry	Netball Manager
17	17/12	Integer	1	Per Entry	Netball Manager

DOCUMENT SPECIFICATION SHEET (INSURANCE REGISTER)

Volumetrics				
Document description	System	Document	Name	Sheet

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

Register	Netball Insurance Register	1	Andrew Burton	1

Stationery ref.	Size	Number of parts	Method of preparation
Register	A4 Landscape	1	Filled in by hand

Filing sequence	Medium	Prepared by
Not filed	Paper	Club Manager

Frequency of preparation	Retention period	Location of file
Once per term	Not retained	Manager's house

	Minimum	Maximum	Av/Abs	Growth rate/fluctuations
Volume	0	1 per session	1 per session	

Users/receipts	Purpose	Frequency of use
Netball Managers	Records who attends and how much they paid.	Once a session, with a session every week.

Data Dictionary

Ref	Name	Data Type	Length	Occurrence	Source of data
1	Name	String	10	Per Entry	Netball Secretary
2	1	Date	8	Per Entry	Netball Manager
3	2	Date	8	Per Entry	Netball Manager
4	3	Date	8	Per Entry	Netball Manager
5	4	Date	8	Per Entry	Netball Manager
6	5	Date	8	Per Entry	Netball Manager
7	6	Date	8	Per Entry	Netball Manager
8	7	Date	8	Per Entry	Netball Manager
9	8	Date	8	Per Entry	Netball Manager
10	9	Date	8	Per Entry	Netball Manager
11	10	Date	8	Per Entry	Netball Manager
12	11	Date	8	Per Entry	Netball Manager
13	12	Date	8	Per Entry	Netball Manager
14	13	Date	8	Per Entry	Netball Manager
15	14	Date	8	Per Entry	Netball Manager
16	15	Date	8	Per Entry	Netball Manager
17	16	Date	8	Per Entry	Netball Manager
18	17	Date	8	Per Entry	Netball Manager
19	18	Date	8	Per Entry	Netball Manager

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

20	19	Date	8	Per Entry	Netball Manager
21	20	Date	8	Per Entry	Netball Manager
22	21	Date	8	Per Entry	Netball Manager

DATA DICTIONARY

The data that I will have to store in the database.

Field Name	Field Purpose	Field Type	Field Size	Example Data	Validation
First Name	Stores the first name of the player	String	30	Owain	Not Blank
Surname	Stores the surname of the player	String	30	Lansdowne	Not Blank
Date Of Birth	Stores the date of birth of the player	Date	YYYY-MM-DD 6 digits	24/09/2020	It's a valid date which was in the past.
Preferred Position	Stores the preferred netball position for the player	String	2	GK	Not blank and is a netball position GK,GD,WD,C,WA,GA,GS
2nd Favourite Position	Stores the second favourite netball position for the player	String	2	GD	Not blank and is a netball position GK,GD,WD,C,WA,GA,GS
3rd Favourite Position	Stores the third favourite netball position for the player	String	2	WD	Not blank and is a netball position GK,GD,WD,C,WA,GA,GS
Email Address	Stores the email address of the player	String	30	189503@ godalming.ac.uk	Not blank and contains a @ sign
Date Of Session	Stores the date of the session	Date	YYYY-MM-DD 6 digits	25/09/2020	That the input is a valid date and in a correct form and in the future
Location of Session	Stores the Location of the session	String	30	Godalming College	Presence Check
Start Time of Session	Stores the start of the session	Time	5 Hour: Minute	20:00	Checks that the start time is before the end time
End Time of Session	Stores the end time of the session	Time	5 Hour: Minute	21:00	Check to ensure that the end time is after the start time

Insurance Start Date	Stores the start date of the persons insurance	Date	YYYY-MM-DD 6 digits	25/03/2020	Validates that a valid date has been entered
Players Position	Stores the position that the player is playing in the session	String	2	GK	A netball position is entered
Players Team	Stores the team that the player is playing for.	String	1	A	Presence check
MVP_Player	Stores the players ID who was best at the session	int	3	2	That input is a valid ID Number

REQUIREMENTS

1. User Input
    i. A player online can add whether they are available to play in one of the sessions which the manager has scheduled.
    ii. A manager's input options are
        a. Add a new player's details
        b. Modify a player's details
        c. Add a session
        d. Add a player's attendance for a session
        e. Remove a player
        f. Generate the teams
        g. Award MVP Player for a session
2. The Processes
    i. Adds a new player's details to the database
    ii. Modifies a player's details on the database
    iii. Adds a player's attendance
    iv. Calculates the amount due by the player for the session
    v. Adds a new session to the database
    vi. Sends the email out to ask for player availability to all the players on the system with details about the session on the email.
    vii. Removes a player from the database
    viii. Saves MVP Players for the session into the database
    ix. Reads the players from the database
    x. Reads the sessions from the database
    xi. Generates an even team when there are 9 or more players taking into consideration the players' preferred positions, the number of times the player has been selected as one of the best players at a session and the number of times a player has been substitute or played their preferred position.
    xii. Saves the team in the database
    xiii. Sends the team out in the database to all the players who have selected that they could play in the session
3. Storage
    i. All data will be saved in the database
    ii. There will be 7 tables

a. Players
b. Sessions
c. Attendance
d. Insurance
e. Player's Preferred Position
f. Player Of The Session
g. Positions

4. Outputs
   i. A new player's ID Number will be displayed
   ii. Error message will be displayed if the inputs are invalid
   iii. An email with the details (Start Time, End Time, Location, Date) will be sent to all the players on the system.
   iv. When the manager has selected to generate the teams, the windows form will display the players on each team and each player's position.
   v. If the player can attend it will display the price the player needs to pay to play.
   VI. The list of teams will be emailed out to all the players who are playing in that session.

DIALOGUE

I need to store the data about each of the players on a database so that the data can be saved when the program is closed. By contrast none of the new data would be saved if all the data were to be declared and set in the program which would mean that the user would constantly need to make changes in the code to save the data. Using a database means that if a player gets insurance then the Secretary can edit the data to save the fact that the user has now got insurance. The advantages of using a database are that the data can be linked together and can be stored effectively and efficiently as well as meaning that it is easy to search and sort through the data about the players. Consequently, you will be able to see how many times a player with insurance has turned up and it will be easy to get certain data from the datasheet such as the preferred position of each of the players attending.

Using a database will allow me to create a web server front end which means that each of the players can be sent the link. The players will be able to submit whether they can attend and the database will be updated so that it records whether they are attending. This would not be possible if all the data was saved in the program as in that situation the Manager would have to enter manually for each of the players whether they would be attending and the players would have had to inform the manager if they were attending. Having a web server front end means that the players don't need access to the program; they only need the link to be able to submit whether they will be attending. In addition the majority of the players are happy to go online and submit whether they are planning to turn up.

For the interface with the Manager I will be using the Windows forms which will mean that they only need to press a button once the players have submitted whether they are attending and then the program will generate the teams based on the players' preferred positions. As a result the players can play their favourite position which will probably be their best position leading to closer, more fun games. I will use the players' MVP points to ensure that the teams are close by ensuring both teams have a roughly even number of players who have played the best at the recent sessions. The other way I will ensure the teams are even is by making sure that the teams have a similar number of players who are playing in their preferred position.

DESIGN

FORMS DESIGN

NAVIGATION DIAGRAM

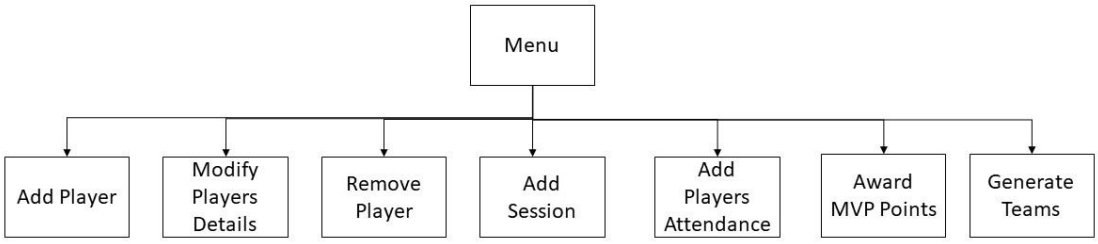


Figure 12 – Forms design

MAIN MENU FORM DESIGN

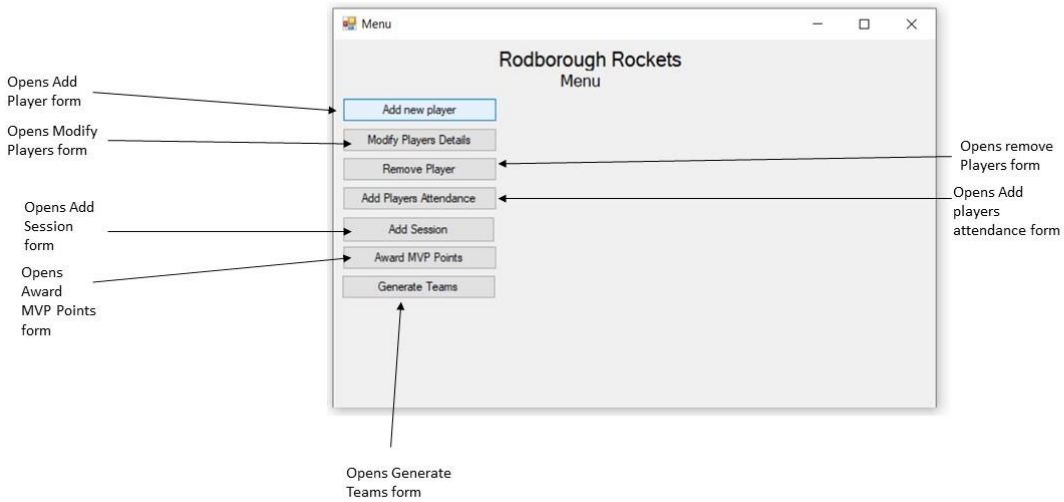


Figure 13 - Menu design

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

ADD PLAYER FORM DESIGN

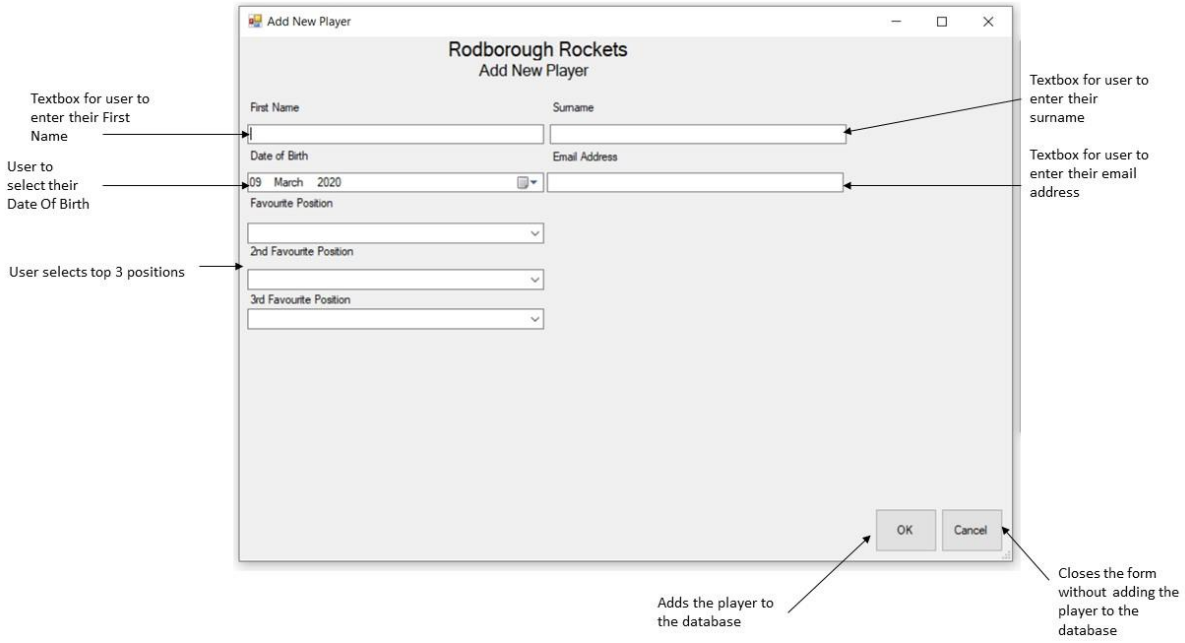


Figure 14 - Add player form design

MODIFY PLAYERS DETAILS FORM DESIGN

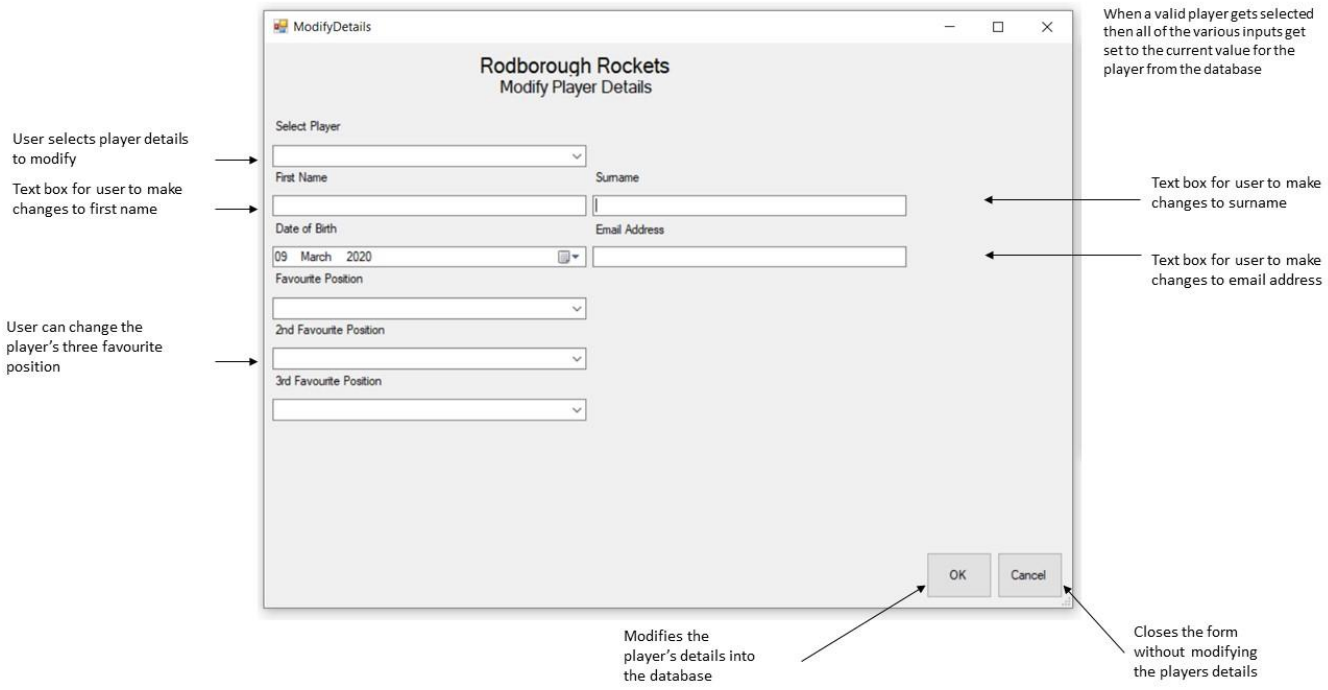


Figure 15 - Modify players details design

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

REMOVE PLAYER FORM DESIGN

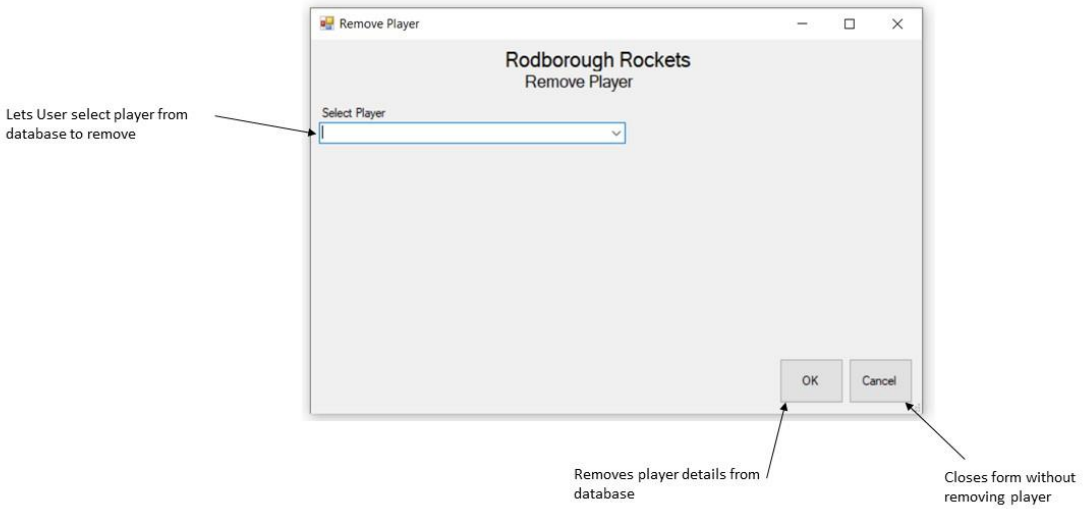


Figure 16 - Remove player form design

ADD SESSION FORM DESIGN

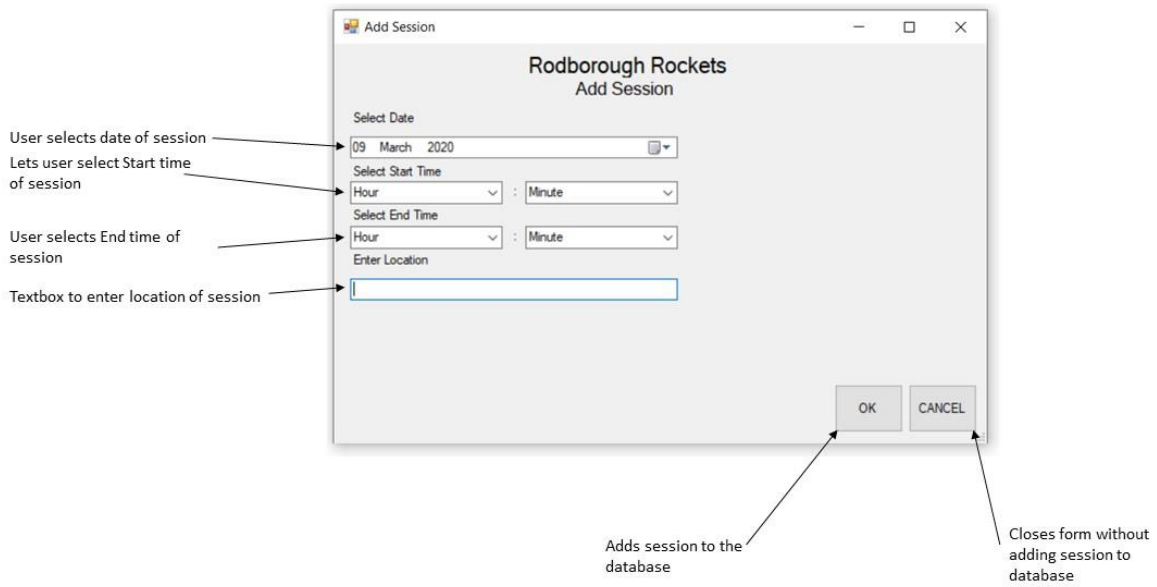


Figure 17 - Add session form design

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

ADD ATTENDANCE FORM DESIGN

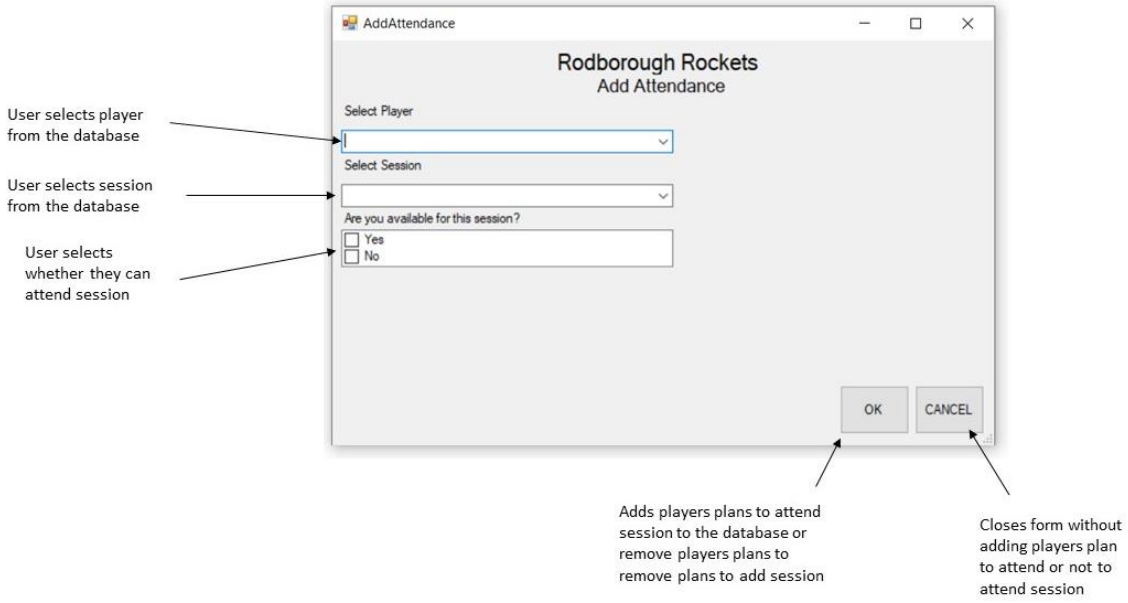
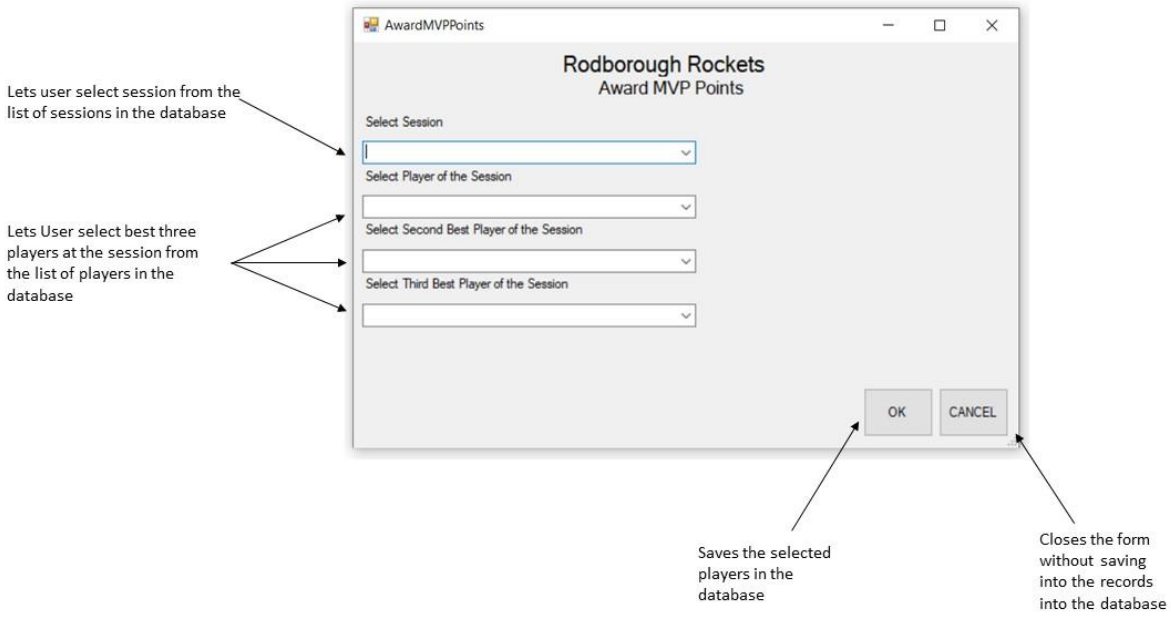


Figure 18 - Add attendance form design

AWARD MVP POINTS DESIGN



Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

Figure 19 - Award MVP Points form design

GENERATE TEAMS FORM DESIGN

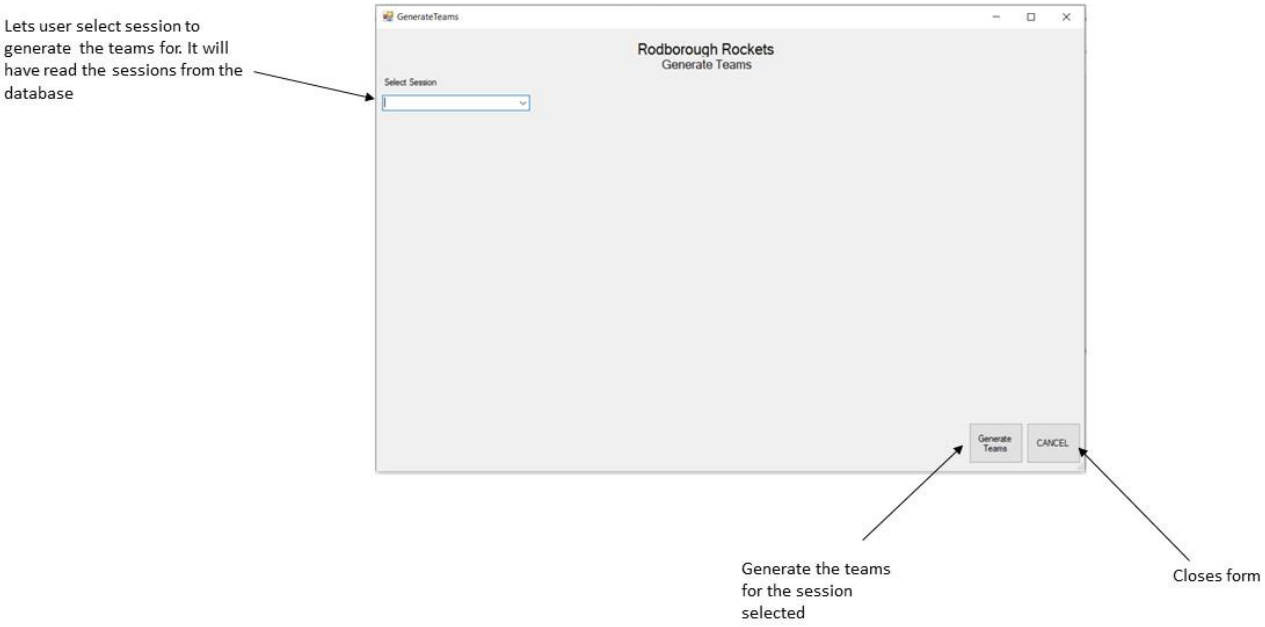


Figure 20 - Generate Teams Form design

VALIDATION

Table with all the validation that will be need in the program.

Field Name	Validation Checks	Description	Error Message
First Name	Presence	Check a name is entered and a suitable length.	Please insert your first name.
Surname	Presence	Check a name is entered and a suitable length.	Please insert your surname.
Email Address	Presence and Type	Checks an email address is entered and it contains an @	Please enter a valid email address
Preferred Position	Lookup (“GK”,”GD”,”WD”,”C”,”WA”, ”GA”,”GS”)	Only allow correctly formatted entries and not the same position as their second favourite position or their third favourite position.	Invalid input, please enter your preferred netball position.

2nd Favourite Position	Lookup (“GK”,”GD”,”WD”,”C”,”WA”, ”GA”,”GS”)	Only allow correctly formatted entries and not the same position as their preferred position or their third favourite position.	Invalid input, please enter your 2nd favourite netball position.
3rd Favourite Position	Lookup (“GK”,”GD”,”WD”,”C”,”WA”, ”GA”,”GS”)	Only allow correctly formatted entries and not the same position as their preferred position or their second favourite position.	Invalid input, please enter your 3rd favourite netball position.
Insurance Choice	Lookup (“YES”,”NO”)	Checks whether they have selected whether they have insurance	Please select whether you have insurance
Player	Lookup from the database	Checks whether they have selected a valid player	Please select a valid player
Session	Lookup from the database	Checks whether they have selected a valid session	Please select a valid session
Session Location	Presence	Check a session is entered and a suitable length.	Please enter a location.
Session Start Time	Presence	Checks a Start Time is entered, and the end time is after start time.	Please enter a valid start time
Session End Time	Presence	Checks an End Time is entered, and the end time is after start time.	Please enter a valid end time.

E-R DIAGRAM

A diagram shows the relationship between the different tables in the database.

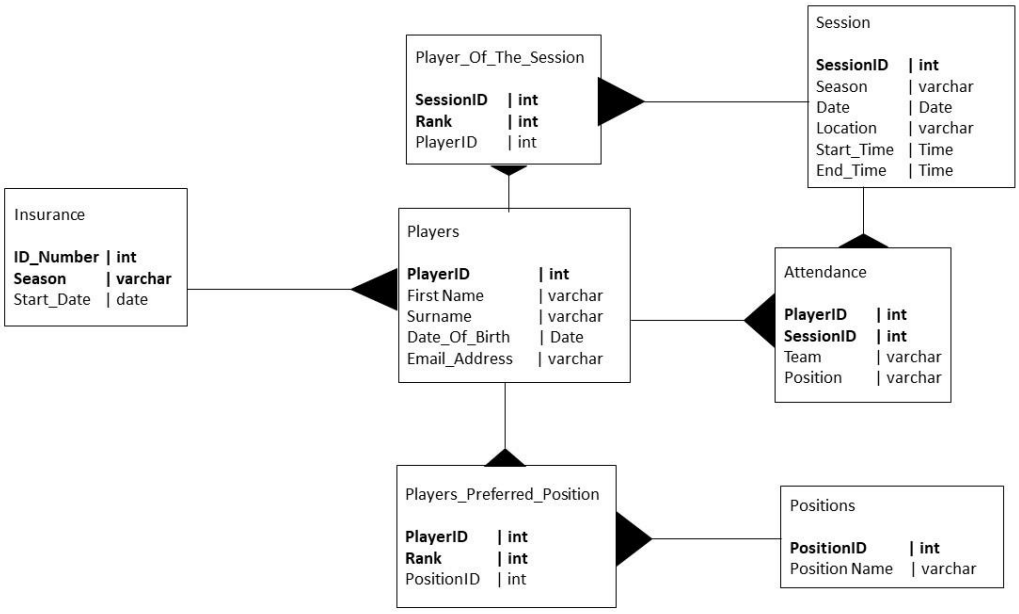


Figure 21 - Entity Relationship Diagram

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

Insurance table and Players table relationship
One player can have many insurances as they can insurance for each season that they are playing.

Player table and Players_Preferred_Position table relationship
One player will have multiple Preferred position as they will be asked for their three-favourite position.

Positions table and Players_Preferred_Position table relationship
One position can be many peoples preferred position.

Attendance table and Player table relationship
One player can have many attendances as they can attend multiple sessions.

Attendance table and Session table relationship
One session can have many attendances as multiple players will be attending each session.

Session table and Player_Of_The_Session table relationship
Each session will have the three best players saved.

Player table and Player_Of_The_Session table relationship
One player can be one of the best players at multiple sessions.

DDL

The Data definition language for each of the tables in the database.

```
CREATE TABLE Players
(PlayerID Int,
FirstName varchar(12),
Surname varchar(12),
Date_Of_Birth date,
Email_Address varchar(20)
PrimaryKey(PlayerID)
);
```

```
CREATE TABLE insurance

(IDNumber INT,
Season varchar(5),
Start_Date date,
PrimaryKey(IDNumber,Season)
);
```

```
CREATE TABLE attendance

(SessionID INT,
PlayerID INT,
team varchar(3),
```

position varchar(3),
PrimaryKey(PlayerID,SessionID)
);

CREATE TABLE session

(SessionID INT,
Season varchar(5),
Date date,
Location varchar(20),
Start_Time time,
End_Time time,
PrimaryKey(SessionID)
);

CREATE TABLE Player_Of_The_Session

(SessionID INT,
PlayerID INT,
Rank INT,
PrimaryKey(SessionID,PlayerID)
);

Create Table Positions

(PositionID INT,
PositionName varchar(2)
PrimaryKey(PositionID)
);

Create Table Players_Preferred_Position

(PlayerID INT,
Rank INT,
PositionID INT
PrimaryKey(PlayerID, Rank)
);

SQL COMMANDS

The SQL commands that I will use in the program.

Selects Number of players on the database to generate PlayerID for new player.

```
SELECT MAX(`ID_Number`) FROM `players`;
```

Inserts new player's details into the database.

```
INSERT INTO `players`(`ID_Number`, `First_Name`, `Surname`, `Date_Of_Birth`, `Email Address`) VALUES (?,?,?,?,?);
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

Inserts new player's favourite positions into the database

```
INSERT INTO `players_preferred_position`(`PlayerID`, `PositionID`, `Rank`) VALUES (?,?,?);
```

Inserts instances of player's attendance into the attendance database.

```
INSERT INTO `attendance` (`PlayerID`, `Date`)
VALUES(? ,? );
```

Inserts instance of insurance for player

```
INSERT INTO `insurance`(`ID_Number`, `Season`, `Start_Date`) VALUES ('?,?,?');
```

Modifies the players details

```
UPDATE `players`
SET `First_Name`= ? ,`Surname`= ? ,`Date_Of_Birth`= ? ,`Email Address`= ?
WHERE `ID_Number` = ?;
```

Modifies the players preferred positions

```
UPDATE `players_preferred_position`
SET `PositionID`=[value-2]
WHERE `PlayerID` = ? and `Rank` = ?
```

Inserts session to the database

```
INSERT INTO `session`(`SessionID`,`Season` ,`Date`, `Location`, `Start_Time`, `End_Time`) VALUES (?,?,?,?,?,?);
```

Reads highest session id number

```
SELECT MAX(`SessionID`) From `session`;
```

Reads the list of players

```
SELECT * FROM `players`;
```

Reads the sessions in the database for player to add attendance

```
SELECT * FROM `session`
WHERE `Date` >= ?
ORDER by `Date` ASC;
```

Inserts a player's attendance if they can attend the session

```
INSERT INTO `attendance`(`SessionID`, `PlayerID`) VALUES (?,?);
```

Removes attendance if they can't attend the session

```
DELETE FROM `attendance` WHERE `SESSIONID` = ? and `PlayerID` = ?;
```

Reads list of players and their details to generate teams

```
SELECT `players`.`First_Name`,`players`.`Surname`,`players`.`Email Address`,`players`.`ID_Number`
From `players`
INNER Join `attendance` on `players`.`ID_Number` = `attendance`.`PlayerID`
Where `attendance`.`SessionID` = ?;
```

Reads player's favourite positions

```
Select `positions`.`Position Name`
From `positions`
INNER JOIN `players_preferred_position` on `positions`.`PositionID` = `players_preferred_position`.`PositionID`
Where `players_preferred_position`.`PlayerID` = ? and `players_preferred_position`.`Rank` = ?;
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

Reads number of sessions a player has attended

```
SELECT Count(*)
From `attendance`
INNER Join `session` on `attendance`.`SessionID` = `session`.`SessionID`
Where `attendance`.`PlayerID` = ? and `session`.`Date` > '2020-10-11';
```

Reads number of times a player has played their preferred position

```
SELECT Count(*)
From `attendance`
INNER Join `session` on `attendance`.`SessionID` = `session`.`SessionID`
Where `attendance`.`PlayerID` = ? and `session`.`Season` ? and `attendance`.`Position` = 'GK'
```

Reads number of times a player has been substitute

```
SELECT Count(*)
From `attendance`
INNER Join `session` on `attendance`.`SessionID` = `session`.`SessionID`
Where `attendance`.`PlayerID` = ? and `session`.`Season` ? and `attendance`.`Position` = 'SUB'
```

Removes a player

```
DELETE FROM `players` WHERE `ID_Number` = ?;
DELETE FROM `insurance` WHERE `ID_Number` = ?;
DELETE FROM `attendance` WHERE `PlayerID` = ?;
DELETE FROM `player_of_the_session` WHERE `PlayerID` = ?;
DELETE FROM `players_preferred_position` WHERE `PlayerID` = ?;
```

Adds player of the session

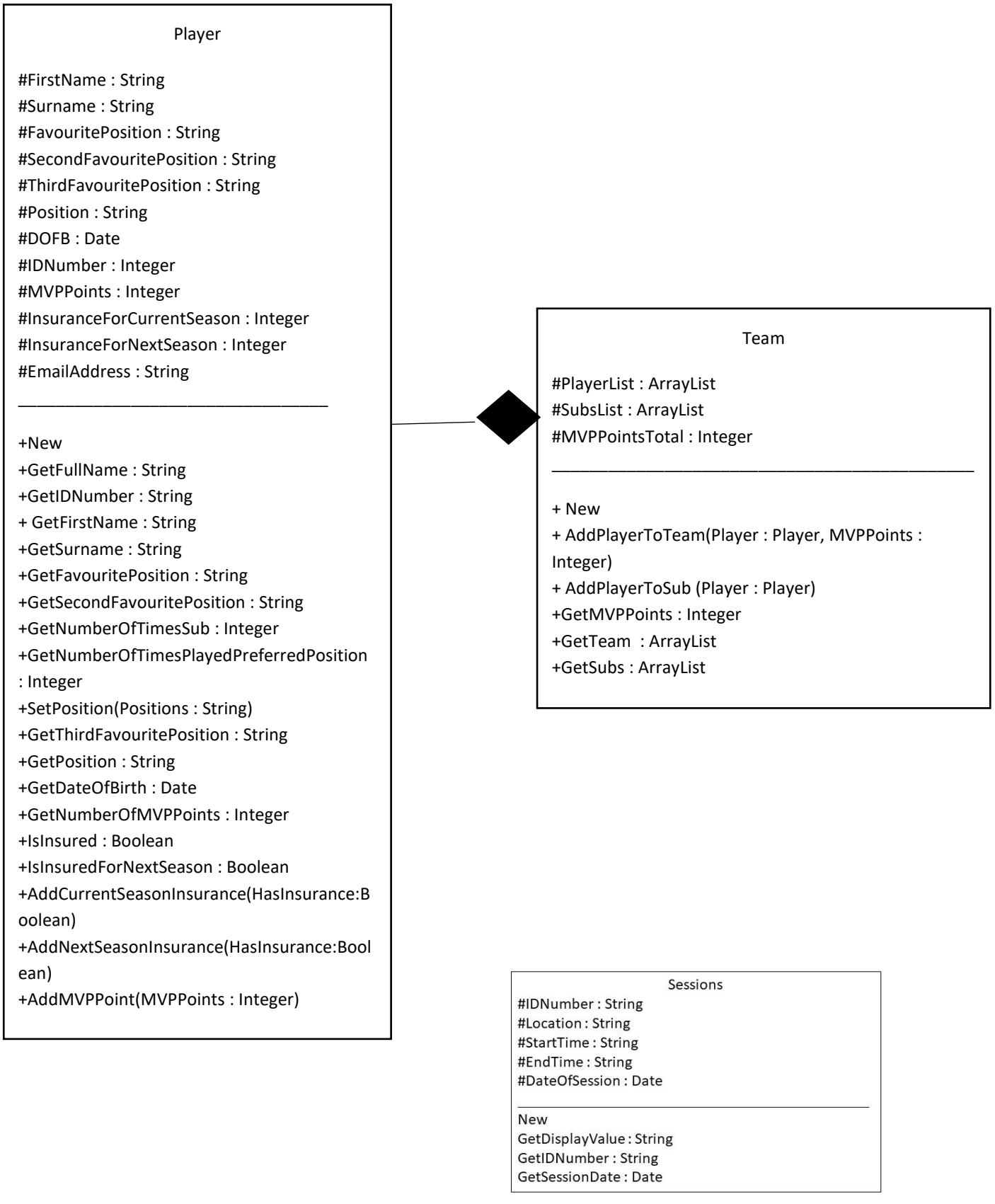
```
INSERT INTO `player_of_the_session`(`SessionID`, `Rank`, `PlayerID`) VALUES (?,?,?);
```

Reads how many times a player has been player of the session

```
SELECT Count(*)
From `player_of_the_session`
INNER JOIN `session` on `player_of_the_session`.`SessionID` = `session`.`SessionID`
Where `session`.`Season` = ? and `Rank` = ? and `player_of_the_session`.`PlayerID` = ?
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

UML CLASS DIAGRAM



Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

DATA DICTIONARY

The data dictionary for the data stored in each of the classes in the program

Player Class

PlayerID will store the ID of the player

Name will store the name of the Player

DateOfBirth will store the date of birth.

Email Address will store the email address of the player

PreferredPosition will store the position that the player prefers.

SecondFavouritePosition stores the position that is the 2nd favourite of the player.

ThirdFavouritePosition stores the position that is the 3rd Favourite of the player.

Position stores the netball position that the player has been allocated.

ModifyDetails will enable user to modify details for the player.

CalculateAmountDue will calculate the amount the player needs to pay in accordance to whether they are an adult and how many weeks they have attended.

GetPreferredPosition will return the position that the player prefer.

GetMVPPoints will return the MVP points the player has.

SetPositons will set the position that the player has been allocated.

Teams Class

Players() will record the details of the players in the team.

Subs() will record the details of the substitutes in the team.

MVPTotal will store the current total in MVP Points in the team.

AddPlayer will add the player to the Players in the array.

ClearTeam will remove all the players in the team.

GetTeam will return the list of players in the team.

SetMVPTotal will add the players' MVP Points to MVP points total of the team.

GetMVPTotal will return the current MVP points total for the team.

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

Session Class

IDNumber will store the ID number of the session

Location will store the location of the session

StartTime will store the start time of the session

EndTime will store the end time of the session

DateOfSession will store the start time of the session

GetDisplayValue will return the Date, Location and Start time as one string to allow the user to select a session

GetIDNumber will return the ID Number of the session

GetSessionDate will return the Date of the session

PSEUDOCODE

Calculates the amount due for the person.

```
Function CalculateAmount(Insured AS Boolean, Adult As Boolean,
NumberOfWeeksAttended AS Integer)
IF Adult = True Then
       If Insured = True THEN
              If NumberOfWeeksAttended =< 21 THEN
                     OUTPUT ‘Amount Due = 2’
              ELSE
                     OUTPUT ‘Amount Due = 1’
       ELSE
              OUTPUT ‘Amount Due = 3’

ELSE
       OUTPUT ‘Amount Due = 1’
```

```
Allocates Players to team when they have been assigned positions
IF TeamA.NumberOfPlayersPlayingPreferredPosition > TeamB.NumberOfPlayersPlayingPreferredPosition THEN
        IF PlayerA.PlayingPreferredPosition = True AND PlayerB.PlayingPreferredPosition = FALSE THEN
                TEAMB.Add(PlayerA)
                TEAMA.Add(PlayerB)
        ELSEIF PlayerA.PlayerPreferredPosition = FALSE and PlayerB.PlayingPreferredPosition = TRUE
                TEAMA.Add(PlayerA)
                TEAMB.Add(PlayerB)
        ELSE
                COMPARE_MVP_Points()
ELSE IF TeamB.NumberOfPlayersPlayingPreferredPosition > TeamA.NumberOfPlayersPlayingPreferredPosition

        IF PlayerA.PlayerPreferredPosition = True AND PlayerB.PlayingPreferredPosition = FALSE THEN
```

```
            TEAMA.Add(PlayerA)
            TEAMB.ADD(PlayerB)
        ELSEIF PlayerA.PlayerPreferredPosition = FALSE and PlayerB.PlayingPreferredPosition = TRUE
            TEAMA.Add(PlayerB)
            TEAMB.Add(PlayerA)
        ELSE
            COMPARE_MVP_Points()
ELSE
        COMPARE_MVP_Points()
```

Compare MVP Points Sub Decides which player should join each team based on the number of MVP Points each team has and the number of MVP Points each player has

```
IF TEAMA.MVPPoints >= TEAMB.MVPPoints THEN
        IF PLAYERA.MVPPoints >= PLAYERB.MVPPoints THEN
            TEAMB.ADD(PLAYERA)
            TEAMA.ADD(PLAYERB)
        ELSE
            TEAMA.ADD(PLAYERA)
            TEAMB.ADD(PLAYERB)
ELSE
        IF PLAYERB.MVPPoints >= PLAYERA.MVPPoints THEN
            TEAMB.ADD(PLAYERA)
            TEAMA.ADD(PLAYERB)
        ELSE
            TEAMA.ADD(PLAYERA)
            TEAMB.ADD(PLAYERB)
```

TESTING PLAN

ALPHA TESTING

Test Number	Description	Data Type	Input	Expected Result
Menu Form				
1	Add Player Button clicked	Typical	Button Clicked	Add Player Form Opens
2	Modify Player Button Clicked	Typical	Button Clicked	Opens Modify Player Form
3	Remove Player Button Clicked	Typical	Button Clicked	Opens Remove Button Form
4	Add Player Attendance Button Clicked	Typical	Button Clicked	Opens Add Attendance Form
5	Add Session Button Clicked	Typical	Button Clicked	Opens Add Session Form
6	Award MVP Points Button Clicked	Typical	Button Clicked	Opens Award MVP Points Form
7	Generate Teams Button Clicked	Typical	Button Clicked	Opens Generate Teams Form
Add Player				
8.1	First name input	Typical	Owain	Accepts the value for the First Name
8.2	First name input	Erroneous		Rejects the value for the First Name
9.1	Surname input	Typical	Lansdowne	Accepts the value for the surname
9.2	Surname input	Erroneous		Rejects the value for the surname
10.1	Preferred Positions Input – The player's favourite position is a valid netball position.	Typical	GK	Accepts the value for the Preferred Positions.
10.2	Preferred Positions Input – The player's favourite position is a valid netball position.	Erroneous	CF	Rejects the value for the Preferred Positions.
11.1	Second Favourite Positions Input – The player's second favourite	Typical	WD	Accepts the value for the Second

	position is a valid netball position.			Favourite Position.
11.2	Second Favourite Positions Input – The player's second favourite position is a valid netball position.	Erroneous	CB	Rejects the value for the Second Favourite Position.
12.1	Third Favourite Positions Input – The player's third favourite position is a valid netball position.	Typical	GA	Accepts the value for the Third Favourite Position.
12.2	Third Favourite Positions Input – The player's third favourite position is a valid netball position.	Erroneous	FB	Rejects the value for the Third Favourite Position.
13.1	The player's three favourite positions are all different	Typical	GA,WD,GK	Accepts the value for the positions
13.2	The player's three favourite positions are all different	Erroneous	GK,GK,GK	Rejects the value for the positions
14.1	The email address is a in valid form.	Typical	189503@ godalming.ac.uk	Accepts the value
14.2	The email address is in a valid form	Erroneous	Netball.Player	Rejects the value
15.1	The player selects whether they have insurance for the current season	Typical	Yes clicked	Displays the start date selector
15.2	The player selects whether they have insurance for the current season	Erroneous	Neither yes or no clicked	Doesn't enter the player to the database and displays error message
16.1	The player selects whether they have insurance for the next season	Typical	Yes clicked	Displays the start date selector
16.2	The player selects whether they have insurance for the next season	Erroneous	Neither yes or no clicked	Doesn't enter the player to the database and displays error message
17	Reads highest ID Number and assigns one above		Button Clicked	Reads Highest ID Number and saves data in the database
18	Saves player's details to database		OK Button clicked	Details saved into database

19	Saves player's insurance details to the database		OK Button Clicked	Insurance details saved into database
20	Saves player's favourite positions to the database		OK Button Clicked	Players favourite position to the database
21	Saves player's next season insurance details to the database		OK Button Clicked	Next Season insurance details saved into the database
22	Form closes when button clicked	Typical	Cancel Button Clicked	Form closes
Modify the players details				
23	Reads all the players' details from the database and displays them in a select box			Displays players' into the select box
24.1	Validates that the player selected is valid	Typical	23 – Player Edited	Shows player's details
24.2	Validates that the player selected is valid	Erroneous	Player	Rejects player and displays error message
25.1	Validates First Name Input	Typical	Player	Accepts the value for the First Name
25.2	Validates First Name Input	Erroneous		Rejects the value for the First Name
26.1	Validates Surname Input	Typical	Has been edited	Accepts the value for the Surname
26.2	Validates Surname Input	Erroneous		Rejects the value for the Surname
27.1	Validates Preferred Netball position is a netball position	Typical	GK	Accepts the value for the Preferred Position.
27.2	Validates Preferred Netball position is a netball position	Erroneous	HA	Rejects the value for the Preferred Position.
28.1	Validates second favourite Netball position is a netball position	Typical	GA	Accepts the value for the second

				favourite Position.
28.2	Validates second favourite Netball position is a netball position	Erroneous	Ha	Rejects the value for the second favourite Position.
29.1	Validates third favourite Netball position is a netball position	Typical	GS	Accepts the value for the third favourite Position.
29.2	Validates third favourite Netball position is a netball position	Erroneous	Ha	Rejects the value for the third favourite Position.
30.1	Validates the three favourite netball positions are different	Typical	WD C WA	Accepts the positions
30.2	Validates the three favourite netball positions are different	Erroneous	C C C	Rejects the positions
31.1	Validates email entered is in a valid form	Typical	189503@ godalming. ac.uk	Accepts Email Address
31.2	Validates email entered is in a valid form	Erroneous	player.netball	Rejects Email Address
32.1	Checks user has selected whether they have insurance	Typical	Yes Selected	Accepts input
32.2	Checks user has selected whether they have insurance	Erroneous	Neither selected	Rejects input
33.1	Checks user has selected whether they have insurance for next season	Typical	Yes Selected	Accepts input
33.2	Checks user has selected whether they have insurance for next season	Erroneous	Neither selected	Rejects input
34	Modifies a player's details in the database	Typical	OK Button Clicked	Modifies details in the database
35	Adds player's insurance details to the database	Typical	OK Button Clicked	Saved details into the database
36	Adds player's next season insurance details to the database	Typical	OK Button Clicked	Saved details into the database
Add Attendance				

37	Reads the list of sessions and adds them to the list box of sessions.			Displays session in list box
38	Reads the list of players and adds them to the list box of player.			Displays players in list box
39.1	Validates the session the user has selected	Typical	07/03/2020 – 20:00 @ Rodborough	Accepts the session selected
39.2	User selects invalid session	Erroneous	Next Session	Rejects the session selected
40.1	Validates the player the user has selected	Typical	1 – Owain Lansdowne	Accepts the player selected
40.1	Validates the player the user has selected	Erroneous	Best Player	Rejects the player selected
41.1	Validates the user input of whether they will be attending the session	Typical	Yes Selected	Accepts input
41.2	Validates the user input of whether they will be attending the session	Erroneous	Neither selected	Rejects input
42	Calculates amount due		The player selected is an adult, has insurance and has attended less than 21 sessions	Displays amount due
43	Adds player's attendance to the database if they can attend the session	Typical	OK Button Clicked	Saves players attendance to the database
44	Removes player's attendance from the database if they can't attend the session	Typical	OK Button Clicked	Removes instance of them attending the session
45	Form closes when button clicked	Typical	Cancel Button Clicked	Form closes
Add Session				
46.1	Validates user's start time input	Typical	Start Time = 15:00	Accepts input
46.2	Validates user's start time input	Erroneous	Hour : Minute	Rejects input and displays error message
47.1	Validates users end time input	Typical	End Time = 16:00	Accepts input
47.2	Validates users end time input	Erroneous	Hour : Minute	Rejects input and displays error message

48.1	Validates start time is before end time	Typical	Start Time = 15:00 End Time = 16:00	Accepts input
48.2	Validates start time is before end time	Erroneous	Start Time = 17:00 End Time = 16:00	Rejects input
49.1	Validates that the user has entered a Location	Typical	Rodborough	Accepts input
49.2	Validates that the user has entered a Location	Erroneous		Rejects input and displays error message
50	Reads the highest ID Number for sessions in the database and adds 1 for the new sessions ID Number	Typical	OK Button Clicked	Displays session ID Number
51	Saves new session to the database	Typical	OK Button Clicked	Adds session to the database
52	Sends email out with details about the session	Typical	Send Reminder Button Clicked	Email sent
Remove Player				
53	Successfully reads list of players from the database	Typical	Remove Player Button Clicked and form loaded	Displays players in a list box
54.1	Validates player to be removed is a player on the list	Typical	18 – Player To Be Removed	Accepts player
54.2	Validates player to be removed is a player on the list	Erroneous	Goalkeeper	Rejects selected player and displays error message
55	Removes player's personal details from database	Typical	OK Button Clicked	Removes details from database
56	Removes player's favourite position details from database	Typical	OK Button Clicked	Removes details from database
Award MVP Points				
57	Reads list of sessions from the database	Typical	Award MVP Points Button clicked, and form opened	Displays session in select box
58.1	Validates session selected is an actual session	Typical	25/02/2020 – 20:00 @ Broadwater	Accepts Value

58.2	Validates session selected is an actual session	Erroneous	Session 1	Rejects value and displays error message
59	Reads list of players who attended the session	Typical	Valid session selected	Displays players in a list box
60.1	Validates that a valid player has been selected	Typical	Owain Lansdowne	Accepts Input
60.2	Validates that a valid player has been selected	Erroneous	Test Player	Rejects Input
61.1	Validates that the same player hasn't been selected 3 times	Typical	Owain Lansdowne Aaron Ramsey Hadleigh Parkes	Accepts input
61.2	Validates that the same player hasn't been selected 3 times	Erroneous	Owain Lansdowne Owain Lansdowne Owain Lansdowne	Rejects input and displays error message
62	Saves the best player of the session into the database	Typical	OK Button Clicked	Saves data into the database
63	Form closes when Cancel Button Clicked	Typical	Cancel Button Clicked	Form Closes
Generate teams				
64	Reads list of sessions from the database	Typical	Form Opened	Displays session in a list box
65.1	Validates that a valid session has been selected	Typical	06/03/2020 - 20:00 @ Rodborough	Accept input
65.2	Validates that a valid session has been selected	Erroneous	Training Session	Rejects input and displays error message
66	Reads list of players from database	Typical	Generate Teams Button Clicked	Teams generated with player who can attend the session from the database
67	Teams generated with 14 players playing	Typical	Generate Teams Button Clicked	Teams are displayed with even statistics
68	Saves teams' details into the database	Typical	Save Teams Buttons Clicked	Teams stored into database
69	Email sent with the teams' details in	Typical	Send Team Button clicked	Teams sent in an email

70	Form closes when Cancel button clicked	Typical	Cancel Button Clicked	Form Closes
71	Fair Teams generated with 14 players playing but with different set of players	Typical	Generate Teams Button Clicked Session 13/03/2020 – 20:00 @ Rodborough	Teams displayed in form
72	Fair Teams generated with 16 players playing	Typical	Generate Teams Button Clicked Session 15/03/2020 – 20:00 @ Rodborough	Teams displayed in form
73	Fair Teams generated with 15 players playing	Typical	Generate Teams Button Clicked Session 14/03/2020 – 20:00 @ Rodborough	Teams displayed in form
74	Fair Teams generated with 13 players playing	Typical	Generate Teams Button Clicked Session 16/03/2020 – 20:00 @ Rodborough	Teams displayed in form
75	Fair teams generated with 12 players playing	Typical	Generate Teams Button Clicked Session 17/03/2020 – 20:00 @ Rodborough	Teams displayed in form
76	Teams generated with 14 players with different set of players	Typical	Generate Teams Button Clicked Session 18/03/2020 – 20:00 @ Rodborough	White Box Testing
77	Teams generated with 11 players playing the session	Typical	Generate Teams Button Clicked Session 19/03/2020 – 20:00 @ Rodborough	Teams displayed in form
Add attendance on online				

71	Reads player from the database	Typical	Web Form Loaded	Add player to the list box
72	Reads session from the database	Typical	Web Form Loaded	Add session to the list box
73.1	Validates users input of whether they will be attending the session	Typical	Yes Selected	Accepts input
73.2	Validates users input of whether they will be attending the session	Erroneous	Nether yes or no selected	Displays Error Message
74	Calculates amount due for the session	Typical	Player selected who is an adult, has insurance and has attended less than 21 sessions.	Out puts Amount Due – 2
75	Saves instance of the player attending the session into the database	Typical	Yes selected and Ok Button clicked	Saves data to the database
76	Removes instance of the player attending the session into the database	Typical	No selected and Ok Button clicked	Removes data from the database

BETA TESTING

Questionnaire

1. Are all the requirements met?
   - ☐ Yes
   - ☐ No

   If you selected No, what requirements haven't been met?

   _____
   _____
   _____
   _____

2. Is the User Interface easy to use?
   - ☐ Yes
   - ☐ No

   Any changes you would made to the user interface?

   _____
   _____
   _____
   _____

3. Have you encountered any problems?
   - ☐ Yes
   - ☐ No

   If you selected Yes, what problems have you encountered?

   _____
   _____
   _____
   _____

4. What did you like about the program?

   _____
   _____
   _____
   _____

5. What improvements would you make to the program?

   _____

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

TECHNICAL SOLUTION

VB FORMS

MENU FORMS

```
1.  Public Class Menu
2.      Private Sub Form1_Load(sender As Object, e As EventArgs)
3.      End Sub
4.      Private Sub BTN_Add_Player_Menu_Click(sender As Object, e As EventArgs) Handles BTN_Add_Player_Menu.Click
5.          Dim AddPlayer As New AddPlayer 'Creates instance of the AddPlayer Form
6.          AddPlayer.ShowDialog() 'Displays the add player form
7.      End Sub
8.      Private Sub BTN_ModifyPlayersDetails_Click(sender As Object, e As EventArgs) Handles BTN_ModifyPlayersDetails.Click
9.          Dim ModifyDetailsForm As New ModifyDetails 'Creates instance of the Modify Details Form
10.         ModifyDetailsForm.ShowDialog() 'Displays the modify details form
11.     End Sub
12.     Private Sub RemovePlayerBTN_Click(sender As Object, e As EventArgs) Handles RemovePlayerBTN.Click
13.         Dim RemovePlayer As New RemovePlayer 'Creates instance of the remove player Form
14.         RemovePlayer.ShowDialog() 'Displays the remove player form
15.     End Sub
16.     Private Sub BTN_Add_Attendance_Click(sender As Object, e As EventArgs) Handles BTN_Add_Attendance.Click
17.         Dim AddAttendance As New AddAttendance 'Creates instance of the add attendance player Form
18.         AddAttendance.ShowDialog() 'Displays the add attendance form
19.     End Sub
20.
21.     Private Sub BTN_AddSession_Click(sender As Object, e As EventArgs) Handles BTN_AddSession.Click
22.         Dim AddSession As New AddSession 'Creates instance of the add session player Form
23.         AddSession.ShowDialog() 'Displays the add session form
24.     End Sub
25.
26.     Private Sub AwardMVPPoints_BTN_Click(sender As Object, e As EventArgs) Handles AwardMVPPoints_BTN.Click
27.         Dim AwardMVPPoints As New AwardMVPPoints 'Creates instance of the remove player Form
28.         AwardMVPPoints.ShowDialog() 'Displays the mvp points form
29.     End Sub
30.     Private Sub BTN_GenerateTeams_Click(sender As Object, e As EventArgs) Handles BTN_GenerateTeams.Click
31.         Dim GenerateTeams As New GenerateTeams '
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
32.         GenerateTeams.ShowDialog()
33.     End Sub
34. End Class
35. Class Player
36.     Protected FirstName, Surname, EmailAddress, FavouritePosition, SecondFavouritePosition, ThirdFavouritePosition, Position As Strin
    g
37.     Protected DOFB As Date
38.     Protected IDNumber, MVPPoints As Integer
39.     Protected InsuranceForCurrentSeason, InsuranceForNextSeason As Boolean
40.     Public Sub New(ByVal IDNumber As Integer, ByVal FirstName As String, ByVal Surname As String, ByVal DOFB As Date, ByVal EmailAddr
    ess As String)
41.         Me.FirstName = FirstName
42.         Me.Surname = Surname
43.         Me.DOFB = DOFB
44.         Me.IDNumber = IDNumber
45.         Me.EmailAddress = EmailAddress
46.     End Sub
47.     Public Sub New(ByVal IDNumber As Integer, ByVal FirstName As String, ByVal Surname As String)
48.         Me.FirstName = FirstName
49.         Me.Surname = Surname
50.         Me.IDNumber = IDNumber
51.     End Sub
52.     Public Sub New(ByVal FirstName As String, ByVal Surname As String, ByVal Position As String, ByVal EmailAddress As String, ByVal
    IDNumber As Integer)
53.         Me.FirstName = FirstName
54.         Me.Surname = Surname
55.         Me.FavouritePosition = Position
56.         Me.IDNumber = IDNumber
57.         Me.EmailAddress = EmailAddress
58.     End Sub
59.     Public Sub New(ByVal EmailAddress As String)
60.         Me.EmailAddress = EmailAddress
61.     End Sub
62.     Public Sub New(ByVal IDNumber As Integer, ByVal FirstName As String, ByVal Surname As String, ByVal FavouritePosition As String)

63.         Me.FirstName = FirstName
64.         Me.Surname = Surname
65.         Me.FavouritePosition = FavouritePosition
66.         Me.IDNumber = IDNumber
67.     End Sub
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
68.     Public Sub New(ByVal IDNumber As Integer, ByVal FirstName As String, ByVal Surname As String, ByVal FavouritePosition As String,
    ByVal SecondFavouritePosition As String, ByVal ThirdFavouritePosition As String)
69.         Me.FirstName = FirstName
70.         Me.Surname = Surname
71.         Me.FavouritePosition = FavouritePosition
72.         Me.SecondFavouritePosition = SecondFavouritePosition
73.         Me.ThirdFavouritePosition = ThirdFavouritePosition
74.         Me.IDNumber = IDNumber
75.     End Sub
76.     Public Sub SavePosition(ByVal SecondFavouritePosition As String, ByVal ThirdFavouritePosition As String)
77.         Me.SecondFavouritePosition = SecondFavouritePosition
78.         Me.ThirdFavouritePosition = ThirdFavouritePosition
79.     End Sub
80.     Public Sub SavePosition(ByVal FavouritePosition As String, ByVal SecondFavouritePosition As String, ByVal ThirdFavouritePosition
    As String)
81.         Me.FavouritePosition = FavouritePosition
82.         Me.SecondFavouritePosition = SecondFavouritePosition
83.         Me.ThirdFavouritePosition = ThirdFavouritePosition
84.     End Sub
85.     'Displays full name
86.     Public Function GetFullName() As String
87.         Dim FullName As String
88.         FullName = FirstName & " " & Surname
89.         Return FullName
90.     End Function
91.     'Returns id number
92.     Public Function GetIDNumber() As String
93.         Return IDNumber
94.     End Function
95.     'Return first name
96.     Public Function GetFirstName() As String
97.         Return FirstName
98.     End Function
99.     'Return surname
100.            Public Function GetSurname() As String
101.                Return Surname
102.            End Function
103.            'Return favourite position
104.            Public Function GetFavouritePosition() As String
105.                Return FavouritePosition
106.            End Function
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
107.            'Return second favourite position
108.            Public Function GetSecondFavouritePosition() As String
109.                Return SecondFavouritePosition
110.            End Function
111.            'Return number of times sub
112.            Public Function GetNumberOfTimesSub() As Integer
113.                Return NumberOfTimesSub(IDNumber, "SUB")
114.            End Function
115.            'Returns number of times player preferred position
116.            Public Function GetNumberOfTimesPlayedPreferredPosition() As Integer
117.                Return NumberOfTimesSub(IDNumber, FavouritePosition)
118.            End Function
119.            'Sets players playing position
120.            Public Sub SetPosition(ByVal Position As String)
121.                Me.Position = Position
122.            End Sub
123.            'Gets third favourite position
124.            Public Function GetThirdFavouritePosition() As String
125.                Return ThirdFavouritePosition
126.            End Function
127.            'Gets players playing position
128.            Public Function GetPosition() As String
129.                Return Position
130.            End Function
131.            'Gets players date of birth
132.            Public Function GetDateOfBirth() As Date
133.                Return DOFB
134.            End Function
135.            'Get number of mvp points
136.            Public Function GetNumberOfMVPPoints() As Integer
137.                Return Me.MVPPoints
138.            End Function
139.            'Returns whether insured
140.            Public Function IsInsured() As Boolean
141.                Return InsuranceForCurrentSeason
142.            End Function
143.            'Returns whether insured for next season
144.            Public Function IsInsuredForNextSeason() As Boolean
145.                Return InsuranceForNextSeason
146.            End Function
147.            'Adds insurance for current season
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
148.            Public Sub AddCurrentSeasonInsurance(ByVal HasInsuranceForCurrentSeason As Boolean)
149.                Me.InsuranceForCurrentSeason = HasInsuranceForCurrentSeason
150.            End Sub
151.            'Adds insurance for next season
152.            Public Sub AddNextSeasonInsurance(ByVal HasInsuranceForNextSeason As Boolean)
153.                Me.InsuranceForNextSeason = HasInsuranceForNextSeason
154.            End Sub
155.            'Adds mvp points
156.            Public Sub AddMVPPoint(ByVal MVPPoints As Integer)
157.                Me.MVPPoints = MVPPoints + Me.MVPPoints
158.            End Sub
159.            'Returns mvp points
160.            Public Function GetMVPPoints() As Integer
161.                Return Me.MVPPoints
162.            End Function
163.            'Returns email address
164.            Public Function GetEmailAddress() As String
165.                Return EmailAddress
166.            End Function
167.        End Class
168.        Module Module1
169.            Dim sConnectionString As String = "server=127.0.0.1;user=root;database=rodborough_rockets;port=3306;password=;"
170.            Dim myConnection As New MySql.Data.MySqlClient.MySqlConnection(sConnectionString)
171.            Function GetWhetherAdult(ByVal Input As Date, ByVal DateToCompare As Date)
172.                Dim DOFBChecker As Date
173.                DOFBChecker = Input
174.                DOFBChecker = DOFBChecker.AddYears(18) 'Adds 18 years on to selected date
175.                If DateToCompare.ToShortDateString >= DOFBChecker Then 'Checks whether the todays date is greater than the Date select
    ed plus 18 years to workout whether they are an adult
176.                    Return True 'If they are an adult returns true
177.                    MsgBox("An Adult")
178.                Else
179.                    Return False 'If They aren't an adult returns false
180.                End If
181.            End Function
182.
183.            Function ConvertDateToSQLDate(ByVal NonSQLDate As Date) As String
184.                'Converts the date in vb to the date in sql date form
185.                Dim DateINSQLForm As String
186.                DateINSQLForm = NonSQLDate.Year & "-" & NonSQLDate.Month & "-" & NonSQLDate.Day
187.                'Returns the date in SQL Form
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
188.                 Return DateINSQLForm
189.             End Function
190.
191.             Function ValidatePositionsInput(ByVal FavouritePosition As String, ByVal SecondFavouritePosition As String, ByVal ThirdFav
    ouritePosition As String) As Boolean
192.                 Dim Positions() As String = {"GK", "GD", "WD", "C", "WA", "GA", "GS"} 'Set of positions in Netball
193.                 Dim FavouritePositionIsNetballPosition, SecondFavouritePositionIsNetballPosition, ThirdFavouritePositionIsNetballPosit
    ion As Boolean 'Boolean variables to check that input for each position is netball position
194.                 FavouritePositionIsNetballPosition = False
195.                 SecondFavouritePositionIsNetballPosition = False
196.                 ThirdFavouritePositionIsNetballPosition = False
197.                 'Error catching
198.                 Try
199.                     'Sets Position variable as the input for the textbox
200.                     If FavouritePosition = SecondFavouritePosition Or FavouritePosition = ThirdFavouritePosition Or SecondFavouritePos
    ition = ThirdFavouritePosition Then 'Checks whether the selected positions aren't the same
201.                         MsgBox("Invalid Position Selection - Can't select the same position multiple times") 'Displays error message
202.                         Return False 'If positions are the same and return False because Position aren't valid inputs
203.                     Else
204.                         If FavouritePosition = "" Or SecondFavouritePosition = "" Or ThirdFavouritePosition = "" Then 'Check a positio
    n has actually been selected
205.                             MsgBox("Invalid Position Selection - You must select a position") 'Displays error message
206.                             Return False 'If position isn't selected return false
207.                         Else
208.                             For Count = 0 To 6 'Loops through each item in the list of positions and checks that FavouritePosition is
    a netball positon
209.                                 If Positions(Count) = FavouritePosition Then
210.                                     FavouritePositionIsNetballPosition = True
211.                                 End If
212.                             Next
213.                             If FavouritePositionIsNetballPosition = False Then 'If favourite position isn't a netball position
214.                                 MsgBox("Favourite Position Selected isn't a netball position") 'Displays message explaining invalid in
    put
215.                                 Return False 'Returns false as input isn't valid
216.                             End If
217.                             For Count = 0 To 6 'Loops through each item in the list of positions and checks that SecondFavouritePositi
    on is a netball positon
218.                                 If Positions(Count) = SecondFavouritePosition Then
219.                                     SecondFavouritePositionIsNetballPosition = True
220.                                 End If
221.                             Next
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                            If SecondFavouritePositionIsNetballPosition = False Then 'If second favourite position isn't a netball pos
    ition
                                MsgBox("Second Favourite Position Selected isn't a netball position") 'Displays message explaining inv
    alid input
                                Return False 'Returns false as input isn't valid
                            End If
                            For Count = 0 To 6 'Loops through each item in the list of positions and checks that ThirdFavouritePositio
    n is a netball positon
                                If Positions(Count) = ThirdFavouritePosition Then
                                    ThirdFavouritePositionIsNetballPosition = True
                                End If
                            Next
                            If ThirdFavouritePositionIsNetballPosition = False Then 'If third favourite position isn't a netball posit
    ion
                                MsgBox("Third Favourite Position Selected isn't a netball position") 'Displays message explaining inva
    lid input
                                Return False 'Returns false as input isn't valid
                            End If
                        End If
                        Return True 'Returns true if each of the players preferred position is valid
                    End If
                Catch ex As Exception
                    MsgBox("Invalid Position Selection") 'Displays error message if not valid
                    Return False 'Returns false as invalid input
                End Try
            End Function
            Function ValidateInsuranceInput(ByVal NetballSelection As String, ByVal HasInsurance As Boolean) As Boolean
                'Validates the players selection of whether they have insurance
                If NetballSelection = "Yes" Or NetballSelection = "No" Or HasInsurance = True Then
                    Return True
                Else
                    'Displays message asking to user to select whether they have insurance
                    MsgBox("You must select whether you have insurance")
                    Return False
                End If
            End Function
            Function ValidNames(ByVal FirstName As String, ByVal Surname As String) As Boolean 'Validates Name inputs
                Dim ValidFirstName, ValidSurname As Boolean
                Try
                    If FirstName = "" Then 'Checks Name been inputed
                        MsgBox("You must enter a First Name") 'Displays message saying name needed to be inputted
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                        ValidFirstName = False
                    Else
                        ValidFirstName = True
                    End If
                Catch ex As Exception
                    MsgBox("Invalid First Name - You must enter a new First Name") 'Displays message if invalid input
                End Try
                Try
                    If Surname = "" Then 'Checks surname been inputed
                        MsgBox("You must enter a Surname") 'Displays message saying name needed to be inputted
                        ValidSurname = False
                    Else
                        ValidSurname = True
                    End If
                Catch ex As Exception
                    MsgBox("Invalid Surname - You must enter a new Surname") 'Displays message if invalid input
                End Try
                Return ValidFirstName And ValidSurname 'Returns the result of And logic operation on ValidFirstName and Surname
            End Function
            Sub CalculatesSeasonInsurance(ByVal DateToday As Date, ByRef Season As String, ByRef NewSeason As String, ByRef CanRegisterInsuranceForNextSeason As Boolean)
                'Calculates the seasons that the player can add insurance for
                'If its August onwards
                If DateToday.Month >= "9" Then
                    'Generates the season
                    Season = Mid((DateToday.Year).ToString, 3, 2) & "/" & (CInt(Mid((DateToday.Year).ToString, 3, 2)) + 1)
                    'Can't register insurance for next season
                    CanRegisterInsuranceForNextSeason = False
                Else
                    'If its July or August then they can add insurance for next season
                    If DateToday.Month = "8" Or DateToday.Month = "7" Then
                        'Generates the season and the new season
                        Season = (CInt(Mid((DateToday.Year).ToString, 3, 2)) - 1) & "/" & Mid((DateToday.Year).ToString, 3, 2)
                        NewSeason = (CInt(Mid((DateToday.Year).ToString, 3, 2))) & "/" & (CInt(Mid((DateToday.Year).ToString, 3, 2))) + 1
                        'Can register insurance for next season
                        CanRegisterInsuranceForNextSeason = True
                    Else
                        'Generates the season
                        Season = ((CInt(Mid(DateToday.Year.ToString, 3, 2))) - 1) & "/" & Mid(DateToday.Year.ToString, 3, 2)
                        'Can't register insurance for next season
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                    CanRegisterInsuranceForNextSeason = False
                End If
            End If
        End Sub
        Function ReadMaxIDNumber() As Integer
            Dim msNumberOfEntries As String = 0
            Dim miIDNumber As Integer
            Dim conn As New MySql.Data.MySqlClient.MySqlConnection
            Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
            Dim dr As MySql.Data.MySqlClient.MySqlDataReader
            conn.ConnectionString = sConnectionString
            Try
                'Opens connection to database
                conn.Open()
                cmd.Connection = conn
                'Sets commant text to read max Player id number in the database
                cmd.CommandText = "SELECT MAX(`ID_Number`) FROM `players`"
                cmd.Prepare()
                'Sets parameters value
                dr = cmd.ExecuteReader()
                While dr.Read
                    msNumberOfEntries = (dr("MAX(`ID_Number`)").ToString)
                End While
                'Closes connection to the database
                conn.Close()
            Catch ex As Exception
                'Displays error message
                MsgBox(ex.ToString)
            End Try
            'Adds one to the highest in the database for the players new id number
            miIDNumber = CInt(msNumberOfEntries) + 1
            'Returns the new id number
            Return miIDNumber
        End Function
        Sub ReadIDNumber(ByRef mISessionIDNumber As Integer)
            Dim dr As MySql.Data.MySqlClient.MySqlDataReader
            Dim mSEntryread As String
            Dim conn As New MySql.Data.MySqlClient.MySqlConnection
            Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
            conn.ConnectionString = sConnectionString
            Try
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
338.                     'Opens database
339.                     conn.Open()
340.                     cmd.Connection = conn
341.                     'Reads the highest SessionID Number in the database
342.                     cmd.CommandText = ("SELECT MAX(`SessionID`) From `session`")
343.                     cmd.Prepare()
344.                     dr = cmd.ExecuteReader()
345.                     While dr.Read
346.                         'Stores the value read from the database
347.                         mSEntryread = (dr("MAX(`SessionID`)")).ToString
348.                         If mSEntryread = "" Then
349.                             mISessionIDNumber = 0
350.                         Else
351.                             mISessionIDNumber = CInt(mSEntryread)
352.                         End If
353.                     End While
354.                     'Closes connection to the database
355.                     conn.Close()
356.                     'Adds 1 to the highest ID Number in the database for the new session id
357.                     mISessionIDNumber = mISessionIDNumber + 1
358.                     'Displays new Session ID Number
359.                     MsgBox("Session ID Number - " & mISessionIDNumber)
360.                     'Fails to read highest ID Number
361.                 Catch ex As Exception
362.                     'Displays error message
363.                     MsgBox(ex.ToString)
364.                 End Try
365.             End Sub
366.             Sub AddSessionToDatabase(ByVal mSStartTime As String, ByVal msEndTime As String, ByVal msSeason As String, ByVal mDDate As
    String, ByVal msLocation As String, ByVal mISessionID As Integer)
367.                 Dim msDateSQL As String
368.                 Dim conn As New MySql.Data.MySqlClient.MySqlConnection
369.                 Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
370.                 conn.ConnectionString = sConnectionString
371.                 msDateSQL = ConvertDateToSQLDate(mDDate)
372.                 Try
373.                     'Opens database
374.                     conn.Open()
375.                     cmd.Connection = conn
376.                     ''Sets command text to add session to the database
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                    cmd.CommandText = ("INSERT INTO `session`(`SessionID`,`Season` ,`Date`, `Location`, `Start_Time`, `End_Time`) VALUES (@SessionID,@Season,@Date,@Location,@StartTime,@EndTime)")
                    cmd.Prepare()
                    'Sets parameters value
                    cmd.Parameters.AddWithValue("@SessionID", mISessionID)
                    cmd.Parameters.AddWithValue("@Season", msSeason)
                    cmd.Parameters.AddWithValue("@Date", mDDate)
                    cmd.Parameters.AddWithValue("@Location", msLocation)
                    cmd.Parameters.AddWithValue("@StartTime", mSStartTime)
                    cmd.Parameters.AddWithValue("@EndTime", msEndTime)
                    'Executes SQL Command
                    cmd.ExecuteNonQuery()
                    'Closes connection to the database
                    conn.Close()
                Catch ex As Exception
                    'Displays error message
                    MsgBox(ex.ToString)
                End Try
            End Sub
            Sub ReadPlayersFromDatabase(ByRef PlayersList As ArrayList)
                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
                Dim dr As MySql.Data.MySqlClient.MySqlDataReader
                Dim FavouritePosition, SecondFavouritePosition, ThirdFavouritePosition As String
                Dim IDNumber As String
                conn.ConnectionString = sConnectionString
                Try
                    'Opens connection between database
                    conn.Open()
                    cmd.Connection = conn
                    'Sets command text to read players from database
                    cmd.CommandText = "SELECT * FROM `players`;"
                    cmd.Prepare()
                    dr = cmd.ExecuteReader()
                    While dr.Read
                        'Creates new instance of player
                        Dim NewPlayer As New Player(CInt(dr("ID_Number")), dr("First_Name"), dr("Surname"), dr("Date_Of_Birth"), dr("Email Address"))
                        'Adds new player to the Array List
                        PlayersList.Add(NewPlayer)
                    End While
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                    'Closes connection to the database
                    conn.Close()
                Catch ex As Exception
                    'Displays error message
                    MsgBox(ex.ToString)
                End Try
                'Loops through every player
                For Count = 0 To PlayersList.Count - 1
                    'Read ID Number for the player
                    IDNumber = PlayersList(Count).GetIDNumber
                    'Reads players favourite position
                    FavouritePosition = ReadPositionPerPlayer(IDNumber, 1)
                    'Reads players second favourite position
                    SecondFavouritePosition = ReadPositionPerPlayer(IDNumber, 2)
                    'Reads players third favourite position
                    ThirdFavouritePosition = ReadPositionPerPlayer(IDNumber, 3)
                    'Stores players favourite position
                    PlayersList(Count).SavePosition(FavouritePosition, SecondFavouritePosition, ThirdFavouritePosition)
                Next
            End Sub
            Function ReadPositionPerPlayer(ByVal PlayerID As String, ByVal Rank As String) As String
                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
                Dim dr As MySql.Data.MySqlClient.MySqlDataReader
                Dim ReadValues As String = ""
                conn.ConnectionString = sConnectionString
                Try
                    'Opens connection to the database
                    conn.Open()
                    cmd.Connection = conn
                    'Reads one of the favourite position for the player
                    cmd.CommandText = ("SELECT `Position Name` From positions INNER Join players_preferred_position on positions.Posit
    ionID = players_preferred_position.PositionID Where players_preferred_position.PlayerID = @PlayerID and players_preferred_position.Ra
    nk = @Rank ")
                    cmd.Prepare()
                    'Sets the parameters
                    cmd.Parameters.AddWithValue("@Rank", Rank)
                    cmd.Parameters.AddWithValue("@PlayerID", PlayerID)
                    'Reads from the database
                    dr = cmd.ExecuteReader()
                    While dr.Read
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
455.                        'Reads the position name
456.                        ReadValues = dr("Position Name")
457.                    End While
458.                    'Closes connection to the database
459.                    conn.Close()
460.                    'Returns position name
461.                    Return ReadValues
462.                    'Error catching
463.                Catch ex As Exception
464.                    'Displays error message
465.                    MsgBox(ex.ToString)
466.                    Return ReadValues
467.                End Try
468.            End Function
469.            Sub DeletePlayerFromDatabase(ByVal mSPlayerToBeRemoved As String)
470.                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
471.                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
472.                Dim ReadValues As String = ""
473.                conn.ConnectionString = sConnectionString
474.                Try
475.                    conn.Open()
476.                    cmd.Connection = conn
477.                    'Deletes player details from the database
478.                    cmd.CommandText = ("DELETE From `players` where `ID_Number` = @PlayerID;")
479.                    cmd.Prepare()
480.                    'Sets the parameters
481.                    cmd.Parameters.AddWithValue("@PlayerID", mSPlayerToBeRemoved)
482.                    cmd.ExecuteNonQuery()
483.                    'Fails to read highest ID Number
484.                    conn.Close()
485.                Catch ex As Exception
486.                    'Displays error message
487.                    MsgBox(ex.ToString)
488.                End Try
489.            End Sub
490.            Sub UpdateDatabase(ByVal PlayerIDNumber As Integer, ByVal FirstName As String, ByVal Surname As String, ByVal FavouritePos
    ition As String, ByVal SecondFavouritePosition As String, ByVal ThirdFavouritePosition As String, ByVal DOFB As Date, ByVal EmailAddr
    ess As String)
491.                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
492.                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
493.                conn.ConnectionString = sConnectionString
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
494.                Dim SQLDofb As String
495.                SQLDofb = ConvertDateToSQLDate(DOFB)
496.                Try
497.                    'Opens connection to the database
498.                    conn.Open()
499.                    cmd.Connection = conn
500.                    'Updates players details in the database
501.                    cmd.CommandText = ("UPDATE `players` SET `First_Name`= @FirstName,`Surname`=@Surname,`Email Address`= @EmailAddres
    s,`Date_Of_Birth`= @DOFB WHERE `ID_Number` = @IDNumber;")
502.                    cmd.Prepare()
503.                    'Sets the parameters
504.                    cmd.Parameters.AddWithValue("@FirstName", FirstName)
505.                    cmd.Parameters.AddWithValue("@Surname", Surname)
506.                    cmd.Parameters.AddWithValue("@EmailAddress", EmailAddress)
507.                    cmd.Parameters.AddWithValue("@DOFB", DOFB)
508.                    cmd.Parameters.AddWithValue("@IDNumber", PlayerIDNumber)
509.                    'Executes SQL Command
510.                    cmd.ExecuteNonQuery()
511.                    'Closes connection
512.                    conn.Close()
513.                Catch ex As Exception
514.                    'Displays error message
515.                    MsgBox(ex.ToString)
516.                End Try
517.            End Sub
518.            Sub UpdatePlayersPosition(ByVal Position As String, ByVal PlayerID As String, ByVal Rank As Integer)
519.                Dim PositionID As Integer
520.                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
521.                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
522.                conn.ConnectionString = sConnectionString
523.                PositionID = ReadPositionID(Position)
524.                Try
525.                    'Opens database
526.                    conn.Open()
527.                    cmd.Connection = conn
528.                    'Updates one of players preferred position in the database
529.                    cmd.CommandText = ("UPDATE `players_preferred_position` SET `PositionID`= @Position where `Rank` = @Rank and `Play
    erID` = @PlayerID;")
530.                    cmd.Prepare()
531.                    'Sets the parameters
532.                    cmd.Parameters.AddWithValue("@Position", PositionID)
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
533.                 cmd.Parameters.AddWithValue("@Rank", Rank)
534.                 cmd.Parameters.AddWithValue("@PlayerID", PlayerID)
535.                 'Executes SQL Command
536.                 cmd.ExecuteNonQuery()
537.                 'Closes database
538.                 conn.Close()
539.             Catch ex As Exception
540.                 'Displays error message
541.                 MsgBox(ex.ToString)
542.             End Try
543.         End Sub
544.         Function ReadWhetherHasInsurance(ByVal miIDNumber As Integer, ByVal mSSeason As String) As Boolean
545.             Dim mINumberOfEntries As Integer
546.             Dim dr As MySql.Data.MySqlClient.MySqlDataReader
547.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
548.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
549.             conn.ConnectionString = sConnectionString
550.             'Error catching
551.             Try
552.                 'Opens connection between the database
553.                 conn.Open()
554.                 cmd.Connection = conn
555.                 'Reads whether player has insurance
556.                 cmd.CommandText = "SELECT COUNT(*) From insurance Where `ID_Number` = @IDNumber and `Season` = @Season;"
557.                 cmd.Prepare()
558.                 'Sets parameters
559.                 cmd.Parameters.AddWithValue("@IDNumber", miIDNumber)
560.                 cmd.Parameters.AddWithValue("@Season", mSSeason)
561.                 'Reads details from the database
562.                 dr = cmd.ExecuteReader()
563.                 While dr.Read
564.                     'Stores number of entries
565.                     mINumberOfEntries = CInt(dr("COUNT(*)"))
566.                 End While
567.                 'Closes connection to the database
568.                 conn.Close()
569.                 'Fails to read highest ID Number
570.             Catch ex As Exception
571.                 MsgBox(ex.ToString)
572.             End Try
573.             'Checks number of entries is either 1 or 0
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
574.                If mINumberOfEntries = "1" Or mINumberOfEntries = "0" Then
575.                    'If one entry then return true
576.                    If mINumberOfEntries = "1" Then
577.                        Return True
578.                        'If 0 entry then return false
579.                    Else
580.                        Return False
581.                    End If
582.                    'If multiple entries error message
583.                Else
584.                    'Displays error message
585.                    MsgBox("Multiple entries")
586.                    'Return false
587.                    Return False
588.                End If
589.            End Function
590.            Sub AddInsuranceToDatabase(ByVal mSPlayerIDNumber As String, ByVal mSSeason As String, ByRef mdStartDate As Date) 'Add Ins
    urance to the database
591.                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
592.                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
593.                Dim msStartDateSQLForm As String
594.                msStartDateSQLForm = ConvertDateToSQLDate(mdStartDate)
595.                conn.ConnectionString = sConnectionString
596.                Try
597.                    'Opens connection to database
598.                    conn.Open()
599.                    cmd.Connection = conn
600.                    'Sets commant text to add insurance details into the database
601.                    cmd.CommandText = "INSERT INTO `insurance`(`ID_Number`, `Season`, `Start_Date`) VALUES (@IDNumber,@Season,@Date)"

602.                    cmd.Prepare()
603.                    'Sets parameters value
604.                    cmd.Parameters.AddWithValue("@IDNumber", mSPlayerIDNumber)
605.                    cmd.Parameters.AddWithValue("@Season", mSSeason)
606.                    cmd.Parameters.AddWithValue("@Date", mdStartDate)
607.                    cmd.ExecuteNonQuery()
608.                    'Closes connection to the database
609.                    conn.Close()
610.                Catch ex As Exception
611.                    'Displays error message
612.                    MsgBox(ex.ToString)
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
613.                End Try
614.            End Sub
615.            Function ValidateEmailAddress(ByVal EmailAddress As String) As Boolean
616.                Dim ValidEmailAddress As Boolean = False
617.                'Checks whether has @ symbol in email addres
618.                If InStr(EmailAddress, "@") = 0 Then
619.                    MsgBox("Please enter a valid email address")
620.                    Return False
621.                Else
622.                    Return True
623.                End If
624.            End Function
625.
626.            Sub ADDPlayerToDatabase(ByVal IDNumber As Integer, ByVal FirstName As String, ByVal Surname As String, ByVal FavouritePosi
    tion As String, ByVal SecondFavouritePosition As String, ByVal ThirdFavouritePosition As String, ByVal DOFB As Date, ByVal EmailAddre
    ss As String) 'Adds players details to database
627.                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
628.                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
629.                Dim SQLDate As String
630.                SQLDate = ConvertDateToSQLDate(DOFB)
631.                conn.ConnectionString = sConnectionString
632.                Try
633.                    'Opens connection to database
634.                    conn.Open()
635.                    cmd.Connection = conn
636.                    'Sets commant text to add players details into the database
637.                    cmd.CommandText = "INSERT INTO `players`(`ID_Number`, `First_Name`, `Surname`, `Date_Of_Birth`,`Email Address`) VA
    LUES (@IDNumber,@FirstName,@Surname,@Date,@EmailAddress)"
638.                    cmd.Prepare()
639.                    'Sets parameters value
640.                    cmd.Parameters.AddWithValue("@IDNumber", IDNumber)
641.                    cmd.Parameters.AddWithValue("@FirstName", FirstName)
642.                    cmd.Parameters.AddWithValue("@Surname", Surname)
643.                    cmd.Parameters.AddWithValue("@Date", SQLDate)
644.                    cmd.Parameters.AddWithValue("@EmailAddress", EmailAddress)
645.                    'Executes SQL Command
646.                    cmd.ExecuteNonQuery()
647.                    'Closes connection to the database
648.                    conn.Close()
649.                Catch ex As Exception
650.                    'Displays error message
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
651.                 MsgBox(ex.ToString)
652.             End Try
653.             'Adds players preferred positions to the database
654.             AddPositionToDatabase(FavouritePosition, 1, IDNumber)
655.             AddPositionToDatabase(SecondFavouritePosition, 2, IDNumber)
656.             AddPositionToDatabase(ThirdFavouritePosition, 3, IDNumber)
657.         End Sub
658.         Sub AddPositionToDatabase(ByVal Position As String, ByVal Rank As String, ByVal PlayerID As String)
659.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
660.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
661.             Dim PositionID As Integer
662.             'Gets position ID Number for players preferred position
663.             PositionID = ReadPositionID(Position)
664.             conn.ConnectionString = sConnectionString
665.             Try
666.                 'Opens connection to database
667.                 conn.Open()
668.                 cmd.Connection = conn
669.                 'Sets commant text to add players favourite positions details into the database
670.                 cmd.CommandText = "INSERT INTO `players_preferred_position`(`PlayerID`, `PositionID`, `Rank`) VALUES (@PlayerID,@P
    ositionID,@Rank)"
671.                 cmd.Prepare()
672.                 'Sets parameters value
673.                 cmd.Parameters.AddWithValue("@PlayerID", PlayerID)
674.                 cmd.Parameters.AddWithValue("@PositionID", PositionID)
675.                 cmd.Parameters.AddWithValue("@Rank", Rank)
676.                 cmd.ExecuteNonQuery()
677.                 'Closes connection to the database
678.                 conn.Close()
679.             Catch ex As Exception
680.                 'Displays error message
681.                 MsgBox(ex.ToString)
682.             End Try
683.         End Sub
684.         Function ReadPositionID(ByVal Position As String) As Integer
685.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
686.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
687.             Dim dr As MySql.Data.MySqlClient.MySqlDataReader
688.             Dim ReadValues As String = ""
689.             conn.ConnectionString = sConnectionString
690.             Try
```

```
691.                    'Opens database
692.                    conn.Open()
693.                    cmd.Connection = conn
694.                    'Reads the position ID for a position
695.                    cmd.CommandText = ("SELECT `PositionID` FROM `positions` WHERE `Position Name` = @Position;")
696.                    cmd.Prepare()
697.                    'Sets the parameters
698.                    cmd.Parameters.AddWithValue("@Position", Position)
699.                    'Reads from database
700.                    dr = cmd.ExecuteReader()
701.                    While dr.Read
702.                        'Stores the Position ID
703.                        ReadValues = dr("PositionID")
704.                    End While
705.                    'Closes databse
706.                    conn.Close()
707.                    Return CInt(ReadValues)
708.                Catch ex As Exception
709.                    Return -1
710.                    'Displays error message
711.                    MsgBox(ex.ToString)
712.                End Try
713.            End Function
714.            Sub ReadSessionsDetailsFromDatabase(ByRef SessionsList As ArrayList, ByVal MsTodaysDate As String)
715.                Dim msSqlDate As String
716.                Dim dr As MySql.Data.MySqlClient.MySqlDataReader
717.                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
718.                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
719.                conn.ConnectionString = sConnectionString
720.                'Converts date to the correct sql format
721.                msSqlDate = ConvertDateToSQLDate(MsTodaysDate)
722.                'Error catching
723.                Try
724.                    'Opens connection to database
725.                    conn.Open()
726.                    cmd.Connection = conn
727.                    'Reads session from database
728.                    cmd.CommandText = ("SELECT * FROM `session` WHERE `Date` > @Date ORDER by `Date` ASC")
729.                    cmd.Prepare()
730.                    'Sets parameters value
731.                    cmd.Parameters.AddWithValue("@Date", msSqlDate)
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
732.                     'Reads from the database
733.                     dr = cmd.ExecuteReader()
734.                     While dr.Read
735.                         'Adds new instance to the session
736.                         Dim NewSession As New Sessions(dr("SessionID"), dr("Location"), dr("Date"), dr("Start_Time").ToString, dr("End
    _Time").ToString)
737.                         SessionsList.Add(NewSession)
738.                     End While
739.                     'Closes connection to the database
740.                     conn.Close()
741.                 Catch ex As Exception
742.                     'Displays error message
743.                     MsgBox(ex.ToString)
744.                 End Try
745.             End Sub
746.             Function ReadWhetherGivenAvailability(ByVal mSPlayerIDNumber As String, ByVal mSSessionID As String) As Integer
747.                 Dim dr As MySql.Data.MySqlClient.MySqlDataReader
748.                 Dim conn As New MySql.Data.MySqlClient.MySqlConnection
749.                 Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
750.                 Dim mINumberOfEntry As Integer
751.                 conn.ConnectionString = sConnectionString
752.                 Try
753.                     'Opens connection to database
754.                     conn.Open()
755.                     cmd.Connection = conn
756.                     'Reads whether player has given their availability
757.                     cmd.CommandText = ("Select Count(*) From `session` INNER Join `attendance` On `session`.`SessionID` = `attendance`
    .`SessionID` Where `session`.`SessionID` = @SessionID and `attendance`.`PlayerID` = @PlayerID;")
758.                     cmd.Prepare()
759.                     'Sets parameters value
760.                     cmd.Parameters.AddWithValue("@SessionID", mSSessionID)
761.                     cmd.Parameters.AddWithValue("@PlayerID", mSPlayerIDNumber)
762.                     'Reads from database
763.                     dr = cmd.ExecuteReader()
764.                     While dr.Read
765.                         'Returns count of number of entries
766.                         mINumberOfEntry = dr("Count(*)")
767.                     End While
768.                     'Closes database
769.                     conn.Close()
770.                 Catch ex As Exception
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
771.                 'Displays error message
772.                 MsgBox(ex.ToString)
773.             End Try
774.             'If there is one entry then returns true else returns false
775.             If mINumberOfEntry = 1 Then
776.                 Return True
777.             Else
778.                 Return False
779.             End If
780.         End Function
781.         Sub RemoveAttendance(ByVal mSSessionIDNumber As String, ByVal mSIDNumber As String)
782.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
783.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
784.             conn.ConnectionString = sConnectionString
785.             Try
786.                 'Opens connection to database
787.                 conn.Open()
788.                 cmd.Connection = conn
789.                 'removes attendance if someone can't attend
790.                 cmd.CommandText = ("DELETE FROM `attendance` WHERE `SESSIONID` = @SessionID and `PlayerID` = @PlayerID;")
791.                 cmd.Prepare()
792.                 'Sets parameters value
793.                 cmd.Parameters.AddWithValue("@SessionID", mSSessionIDNumber)
794.                 cmd.Parameters.AddWithValue("@PlayerID", mSIDNumber)
795.                 'Executes SQL Command
796.                 cmd.ExecuteNonQuery()
797.                 'Closes connection to the database
798.                 conn.Close()
799.             Catch ex As Exception
800.                 'Displays error message
801.                 MsgBox(ex.ToString)
802.             End Try
803.         End Sub
804.         Sub AddAttendanceToDatabase(ByVal mSSessionIDNumber As String, ByVal mSIDNumber As String)
805.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
806.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
807.             conn.ConnectionString = sConnectionString
808.             Try
809.                 'Opens connection to database
810.                 conn.Open()
811.                 cmd.Connection = conn
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
812.                    'Adds players attendance
813.                    cmd.CommandText = ("INSERT INTO `attendance`(`SessionID`, `PlayerID`) VALUES (@SessionID,@PlayerID);")
814.                    cmd.Prepare()
815.                    'Sets parameters value
816.                    cmd.Parameters.AddWithValue("@SessionID", mSSessionIDNumber)
817.                    cmd.Parameters.AddWithValue("@PlayerID", mSIDNumber)
818.                    'Executes SQL Command
819.                    cmd.ExecuteNonQuery()
820.                    'Closes connection to the database
821.                    conn.Close()
822.                Catch ex As Exception
823.                    'Displays error message
824.                    MsgBox(ex.ToString)
825.                End Try
826.            End Sub
827.            Sub ReadDatabase(ByVal mSPlayerIDNumber As String, ByRef DateOfBirth As Date)
828.                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
829.                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
830.                Dim dr As MySql.Data.MySqlClient.MySqlDataReader
831.                conn.ConnectionString = sConnectionString
832.                Try
833.                    'Opens database
834.                    conn.Open()
835.                    cmd.Connection = conn
836.                    'Reads players date of birth from database
837.                    cmd.CommandText = ("SELECT`Date_Of_Birth` From `players` where `ID_Number` = @IDNumber;")
838.                    cmd.Prepare()
839.                    'Sets the parameters
840.                    cmd.Parameters.AddWithValue("@IDNumber", mSPlayerIDNumber)
841.                    'Executes SQL Command
842.                    dr = cmd.ExecuteReader()
843.                    While dr.Read
844.                        'Saves date of birth read from the database
845.                        DateOfBirth = dr("Date_Of_Birth")
846.                    End While
847.                    'Closes connection to the database
848.                    conn.Close()
849.                Catch ex As Exception
850.                    'Displays error message
851.                    MsgBox(ex.ToString)
852.                End Try
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
853.            End Sub
854.            Function CalculateAmountDue(ByVal mSPlayerIDNumber As String, ByVal mdDateOfSession As Date, ByVal mdDofB As Date) As Inte
    ger
855.                Dim msSeason, mSInsuranceStartDate As String
856.                Dim mbIsAnAdult, mbHasInsurance As Boolean
857.                Dim miNumberOfAttendances As Integer
858.                'If they are session is september, october or november
859.                If mdDateOfSession.Month >= 9 Then
860.                    'Sets the season for the session
861.                    msSeason = Mid(mdDateOfSession.Year, 3, 2) & "/" & Mid(mdDateOfSession.Year + 1, 3, 2)
862.                Else
863.                    'Sets the season for the session
864.                    msSeason = (Mid(mdDateOfSession.Year - 1, 3, 2) & "/" & Mid(mdDateOfSession.Year, 3, 2))
865.                End If
866.                'Checks using date of birth whether they are an adult
867.                mbIsAnAdult = GetWhetherAdult(mdDofB, mdDateOfSession)
868.                If mbIsAnAdult Then
869.                    'Checks whether they have insurance and reads thier insurance start date
870.                    mbHasInsurance = HasInsurance(msSeason, mSPlayerIDNumber, mSInsuranceStartDate)
871.                    'If they have insurance checks number of sessions they have attended
872.                    If mbHasInsurance Then
873.                        miNumberOfAttendances = CountsNumberOfAttendances(mdDateOfSession, mSPlayerIDNumber, msSeason)
874.                        'If they have attended less then 21
875.                        If miNumberOfAttendances < 21 Then
876.                            'Need to pay 2
877.                            Return 2
878.                        Else
879.                            'Need to pay 3
880.                            Return 3
881.                        End If
882.                    Else
883.                        'They don't have insurance need to pay 3
884.                        Return 3
885.                    End If
886.                Else
887.                    'if They are a child then they need to pay 1
888.                    Return 1
889.                End If
890.            End Function
891.            'Reads whether player has insurance for the season
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
892.            Function HasInsurance(ByVal msSeason As String, ByVal msIDNumber As String, ByRef msInsuranceStartDate As String) As Boole
    an
893.                Dim dr As MySql.Data.MySqlClient.MySqlDataReader
894.                Dim mISelectCount As Integer
895.                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
896.                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
897.                conn.ConnectionString = sConnectionString
898.                Try
899.                    'Opens connection to database
900.                    conn.Open()
901.                    cmd.Connection = conn
902.                    'Reads whether they have insurance for the season and the start date
903.                    cmd.CommandText = ("SELECT COUNT(*) , `Start_Date` From `insurance` Where `Season` = @Season and `ID_Number` = @ID
    Number;")
904.                    cmd.Prepare()
905.                    'Sets parameters value
906.                    cmd.Parameters.AddWithValue("@IDNumber", msIDNumber)
907.                    cmd.Parameters.AddWithValue("@Season", msSeason)
908.                    'Reads from the database
909.                    dr = cmd.ExecuteReader()
910.                    While dr.Read
911.                        'Stores the number of entries and the start date
912.                        mISelectCount = dr("COUNT(*)").ToString
913.                        msInsuranceStartDate = dr("Start_Date").ToString
914.                    End While
915.                    'Closes connection to the database
916.                    conn.Close()
917.                Catch ex As Exception
918.                    MsgBox(ex.ToString)
919.                End Try
920.                'Returns whether they have insurance if there is one entry in the database
921.                If mISelectCount = 1 Then
922.                    Return True
923.                Else
924.                    Return False
925.                End If
926.            End Function
927.            'Counts number of times a player has attended a session
928.            Function CountsNumberOfAttendances(ByVal mdDate As Date, ByVal msIDNumber As String, ByVal msSeason As String)
929.                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
930.                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
931.                Dim dr As MySql.Data.MySqlClient.MySqlDataReader
932.                Dim mISelectCount As Integer
933.                Dim msSQLSessionDate As String
934.                conn.ConnectionString = sConnectionString
935.                'Converts date to sql form
936.                msSQLSessionDate = ConvertDateToSQLDate(mdDate)
937.                Try
938.                    'Opens connection to database
939.                    conn.Open()
940.                    cmd.Connection = conn
941.                    'counts number of time they have attended
942.                    cmd.CommandText = ("SELECT Count(*) From `attendance` INNER JOIN `session` on `session`.`SessionID` = `attendance`
    .`SessionID` Where `session`.`Date` <= @Date and `PlayerID` = @PlayerID and `session`.`Season`= @Season ;")
943.                    cmd.Prepare()
944.                    'Sets parameters value
945.                    cmd.Parameters.AddWithValue("@Date", mdDate)
946.                    cmd.Parameters.AddWithValue("@PlayerID", msIDNumber)
947.                    cmd.Parameters.AddWithValue("@Season", msSeason)
948.                    'Reads from the database
949.                    dr = cmd.ExecuteReader()
950.                    While dr.Read
951.                        mISelectCount = dr("COUNT(*)")
952.                    End While
953.                    'Closes database
954.                    conn.Close()
955.                Catch ex As Exception
956.                    MsgBox(ex.ToString)
957.                End Try
958.                'Returns number of session attended
959.                Return mISelectCount
960.            End Function
961.            Sub ReadListOfSessions(ByRef malListOfSessions As ArrayList, ByVal mdDateToday As Date)
962.                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
963.                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
964.                Dim msSQLSessionDate As String
965.                Dim ListOfSessions As New ArrayList
966.                Dim dr As MySql.Data.MySqlClient.MySqlDataReader
967.                'Converts date to sql form
968.                msSQLSessionDate = ConvertDateToSQLDate(mdDateToday)
969.                conn.ConnectionString = sConnectionString
970.                Try
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                    'Opens connection to database
                    conn.Open()
                    cmd.Connection = conn
                    'Reads list of sessions
                    cmd.CommandText = ("SELECT `Date`,`Location` ,`Start_Time`,`End_Time`,`SessionID`From `session` Where `Date` < @Se
    ssionDate ;")
                    cmd.Prepare()
                    'Sets parameters value
                    cmd.Parameters.AddWithValue("@SessionDate", msSQLSessionDate)
                    dr = cmd.ExecuteReader()
                    While dr.Read
                        'Creates new instance of the session
                        Dim NewSession As New Sessions(dr("SessionID"), dr("Location"), dr("Date"), dr("Start_Time").ToString, dr("End
    _Time").ToString)
                        'Adds new session to the list of sessions
                        ListOfSessions.Add(NewSession)
                    End While
                    'Closes database
                    conn.Close()
                Catch ex As Exception
                    'Displays error message
                    MsgBox(ex.ToString)
                End Try
                'Loops through each session and checks if mvp players have already been awarded for that session
                For Count = 0 To ListOfSessions.Count - 1
                    ReadListOfSessions(malListOfSessions, ListOfSessions(Count).GetIDNumber, ListOfSessions(Count))
                Next
            End Sub
            Sub ReadListOfSessions(ByRef malListOfSessions As ArrayList, ByRef IDNumber As String, ByVal session As Sessions)
                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
                Dim dr As MySql.Data.MySqlClient.MySqlDataReader
                Dim ReadValue As Integer = 0
                conn.ConnectionString = sConnectionString
                Try
                    'Opens database
                    conn.Open()
                    cmd.Connection = conn
                    'Finds out whether session has been awarded mvp points
                    cmd.CommandText = ("SELECT Count(*) From `player_of_the_session` Where `SessionID` = @IDNumber ;")
                    cmd.Prepare()
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1010.                   'Sets the parameters
1011.                   cmd.Parameters.AddWithValue("@IDNumber", IDNumber)
1012.                   'Reads from the database
1013.                   dr = cmd.ExecuteReader()
1014.                   While dr.Read
1015.                       ReadValue = CInt(dr("Count(*)"))
1016.                   End While
1017.                   'Closes connection with database
1018.                   conn.Close()
1019.               Catch ex As Exception
1020.                   'Displays error message
1021.                   MsgBox(ex.ToString)
1022.               End Try
1023.               'If mvp players haven't been awarded then the session is added to the list
1024.               If ReadValue = 0 Then
1025.                   malListOfSessions.Add(session)
1026.               End If
1027.           End Sub
1028.           Sub ReadListOfPlayers(ByRef malListOfPlayers As ArrayList, ByVal msSessionID As String)
1029.               Dim dr As MySql.Data.MySqlClient.MySqlDataReader
1030.               Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1031.               Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1032.               conn.ConnectionString = sConnectionString
1033.               Try
1034.                   'Opens database
1035.                   conn.Open()
1036.                   cmd.Connection = conn
1037.                   'Reads list of players who can attended the session
1038.                   cmd.CommandText = ("Select `players`.`ID_Number`,`players`.`First_Name`,`players`.`Surname` From `players` INNER J
    oin `attendance` On `players`.`ID_Number` = `attendance`.`PlayerID` Where `attendance`.`SessionID` = @SessionID;")
1039.                   cmd.Prepare()
1040.                   'Sets parameters value
1041.                   cmd.Parameters.AddWithValue("@SessionID", msSessionID)
1042.                   dr = cmd.ExecuteReader()
1043.                   While dr.Read
1044.                       'Creates new instance of players
1045.                       Dim NewPlayer As New Player(dr("ID_Number"), dr("First_Name"), dr("Surname"))
1046.                       'Adds player to the list
1047.                       malListOfPlayers.Add(NewPlayer)
1048.                   End While
1049.                   'Closes connection to the database
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1050.                 conn.Close()
1051.             Catch ex As Exception
1052.                 'Displays error message
1053.                 MsgBox(ex.ToString)
1054.             End Try
1055.         End Sub
1056.         Sub AddMVPPointsToDatabase(ByVal PlayerID As String, ByVal msSessionID As String, ByVal rank As String)
1057.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1058.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1059.             conn.ConnectionString = sConnectionString
1060.             Try
1061.                 'Opens database
1062.                 conn.Open()
1063.                 cmd.Connection = conn
1064.                 'Inserts into player of the session the players
1065.                 cmd.CommandText = ("INSERT INTO `player_of_the_session`(`SessionID`, `Rank`, `PlayerID`) VALUES (@SessionID,@Rank,
    @PlayerID);")
1066.                 cmd.Prepare()
1067.                 'Sets the parameters
1068.                 cmd.Parameters.AddWithValue("@SessionID", msSessionID)
1069.                 cmd.Parameters.AddWithValue("@Rank", rank)
1070.                 cmd.Parameters.AddWithValue("@PlayerID", PlayerID)
1071.                 'Executes SQL Command
1072.                 cmd.ExecuteNonQuery()
1073.                 'Closes connection to the database
1074.                 conn.Close()
1075.                 'Error catching
1076.             Catch ex As Exception
1077.                 'Displays Error Message
1078.                 Console.WriteLine(ex.ToString)
1079.             End Try
1080.         End Sub
1081.
1082.         Sub ReadListOfSessionsToGenerateTeams(ByVal mdDateToday As Date, ByRef MALSessionsList As ArrayList)
1083.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1084.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1085.             Dim dr As MySql.Data.MySqlClient.MySqlDataReader
1086.             Dim msDateToday As String
1087.             msDateToday = ConvertDateToSQLDate(mdDateToday)
1088.             conn.ConnectionString = sConnectionString
1089.             Try
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1090.                   'Opens connection to database
1091.                   conn.Open()
1092.                   cmd.Connection = conn
1093.                   'Reads list of session which are coming up
1094.                   cmd.CommandText = ("SELECT `SessionID`,`Date`,`Location`,`Start_Time`,`Season` From `session` Where `Date` >= @Dat
    e ORDER BY `Date` ASC,`Start_Time` ASC;")
1095.                   cmd.Prepare()
1096.                   'Sets parameters
1097.                   cmd.Parameters.AddWithValue("@Date", mdDateToday)
1098.                   'Reads from the database
1099.                   dr = cmd.ExecuteReader()
1100.                   While dr.Read
1101.                       'Creates new instance of session
1102.                       Dim NewSession As New Sessions(dr("SessionID"), dr("Location"), dr("Season"), dr("Date"), dr("Start_Time").ToS
    tring, 1)
1103.                       'Adds new session to the list
1104.                       MALSessionsList.Add(NewSession)
1105.                   End While
1106.                   'Closes database
1107.                   conn.Close()
1108.               Catch ex As Exception
1109.                   'Displays error message
1110.                   MsgBox(ex.ToString)
1111.               End Try
1112.           End Sub
1113.           Sub ReadListOfPlayers(ByRef MALPlayersList As ArrayList, ByVal miSessionID As Integer)
1114.               Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1115.               Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1116.               Dim dr As MySql.Data.MySqlClient.MySqlDataReader
1117.               Dim IDNumber As String
1118.               Dim Positions(3) As String
1119.               conn.ConnectionString = sConnectionString
1120.               Try
1121.                   'Opens connection to database
1122.                   conn.Open()
1123.                   cmd.Connection = conn
1124.                   'Reads list of players who can attend the session and their favourite position
1125.                   cmd.CommandText = ("Select `players`.`ID_Number`,`players`.`First_Name`,`players`.`Surname`,`positions`.`Position
    Name`,`players`.`Email Address` From `players` Inner Join `players_preferred_position` on `players`.`ID_Number`=`players_preferred_po
    sition`.`PlayerID` INNER Join `positions` on `players_preferred_position`.`PositionID` = `positions`.`PositionID` INNER Join `attenda
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
    nce` on `players`.`ID_Number` = `attendance`.`PlayerID` Where `players_preferred_position`.`Rank` = '1' and `attendance`.`SessionID`
    = @SessionID;")
1126.                   cmd.Prepare()
1127.                   'Sets parameters value
1128.                   cmd.Parameters.AddWithValue("@SessionID", miSessionID)
1129.                   'Reads from the database
1130.                   dr = cmd.ExecuteReader()
1131.                   While dr.Read
1132.                       'Creates instance of player
1133.                       Dim NewPlayer As New Player(dr("First_Name"), dr("Surname"), dr("Position Name"), dr("Email Address"), CInt(dr
    ("ID_Number")))
1134.                       'Adds player to the list
1135.                       MALPlayersList.Add(NewPlayer)
1136.                   End While
1137.                   'Closes database
1138.                   conn.Close()
1139.               Catch ex As Exception
1140.                   'Displays error message
1141.                   MsgBox(ex.ToString)
1142.               End Try
1143.               'Loops through every player
1144.               For Count = 0 To MALPlayersList.Count - 1
1145.                   'Finds player id number
1146.                   IDNumber = MALPlayersList(Count).GetIDNumber
1147.                   'Finds players second and third favourite position
1148.                   For index = 2 To 3
1149.                       ReadEachPlayersPosition(MALPlayersList, IDNumber, Positions(index), index)
1150.                   Next
1151.                   'Adds positions to the player
1152.                   MALPlayersList(Count).SavePosition(Positions(2), Positions(3))
1153.               Next
1154.           End Sub
1155.           Sub ReadEachPlayersPosition(ByVal MALPlayersList As ArrayList, ByVal IDNumber As String, ByRef FavouritePositions As Strin
    g, ByVal Rank As String)
1156.               Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1157.               Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1158.               Dim dr As MySql.Data.MySqlClient.MySqlDataReader
1159.               conn.ConnectionString = sConnectionString
1160.               Try
1161.                   'Opens connection to database
1162.                   conn.Open()
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1163.                 cmd.Connection = conn
1164.                 'Reads position name which is one of the players favourite position
1165.                 cmd.CommandText = ("SELECT `Position Name` From `positions` Inner Join `players_preferred_position` on `positions`
    .`PositionID` = `players_preferred_position`.`PositionID` Where `players_preferred_position`.`Rank` = @Rank and `players_preferred_po
    sition`.`PlayerID` = @PlayerID;")
1166.                 cmd.Prepare()
1167.                 'Sets parameter
1168.                 cmd.Parameters.AddWithValue("@Rank", Rank)
1169.                 cmd.Parameters.AddWithValue("@PlayerID", IDNumber)
1170.                 'Executes SQL command
1171.                 dr = cmd.ExecuteReader()
1172.                 While dr.Read
1173.                     'Saves the position namee
1174.                     FavouritePositions = dr("Position Name")
1175.                 End While
1176.                 'Closes database
1177.                 conn.Close()
1178.             Catch ex As Exception
1179.                 'Displays error message
1180.                 MsgBox(ex.ToString)
1181.             End Try
1182.         End Sub
1183.         Function NumberOfTimesSub(ByVal IDNumber As Integer, ByVal Position As String) As Integer
1184.             Dim dr As MySql.Data.MySqlClient.MySqlDataReader
1185.             Dim NumberOfTimesRead As String = 0
1186.             Dim DateToSearch As Date
1187.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1188.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1189.             conn.ConnectionString = sConnectionString
1190.             'Finds date three month earlier
1191.             DateToSearch = Date.Today.AddMonths(-3)
1192.             Try
1193.                 'Opens database
1194.                 conn.Open()
1195.                 cmd.Connection = conn
1196.                 'Counts number of times sub in the last 3 month
1197.                 cmd.CommandText = ("SELECT Count(*)From `attendance`INNER JOIN `session` on `session`.`SessionID` = `attendance`.`
    SessionID`Where `PlayerID` = @PlayerID and `session`.`Date` > @Date and`Position` = @Position;")
1198.                 cmd.Prepare()
1199.                 'Sets parameters value
1200.                 cmd.Parameters.AddWithValue("@PlayerID", IDNumber)
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                cmd.Parameters.AddWithValue("@Date", DateToSearch)
                cmd.Parameters.AddWithValue("@Position", Position)
                dr = cmd.ExecuteReader()
                While dr.Read
                    'Saves number of entries
                    NumberOfTimesRead = dr("Count(*)")
                End While
                'Closes database
                conn.Close()
            Catch ex As Exception
                'Displays error message
                MsgBox(ex.ToString)
            End Try
            'Returns number of times sub
            Return NumberOfTimesRead
        End Function
        Sub SavePlayersTeamDetails(ByVal IDNumber As String, ByVal Team As String, ByVal Position As String, ByVal SessionID As St
ring)
            Dim conn As New MySql.Data.MySqlClient.MySqlConnection
            Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
            conn.ConnectionString = sConnectionString
            Try
                'Opens connection to database
                conn.Open()
                cmd.Connection = conn
                'Saves teams details into the database
                cmd.CommandText = ("UPDATE `attendance` SET `Team`= @Team ,`Position`= @Position WHERE `PlayerID` = @PlayerID AND
 `SessionID` = @SessionID;")
                cmd.Prepare()
                'Sets parameters value
                cmd.Parameters.AddWithValue("@Team", Team)
                cmd.Parameters.AddWithValue("@PlayerID", IDNumber)
                cmd.Parameters.AddWithValue("@Position", Position)
                cmd.Parameters.AddWithValue("@SessionID", SessionID)
                'Closes database
                conn.Close()
                'Error catching
            Catch ex As Exception
                'Displays error message
                Console.WriteLine(ex.ToString)
            End Try
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1240.          End Sub
1241.          Sub RemoveAllAttendancesFromDatabase(ByVal IDNumber As String)
1242.              Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1243.              Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1244.              conn.ConnectionString = sConnectionString
1245.              Try
1246.                  'Opens database
1247.                  conn.Open()
1248.                  cmd.Connection = conn
1249.                  'removes players attendance details
1250.                  cmd.CommandText = ("DELETE FROM `attendance` WHERE `PlayerID` = @PlayerID;")
1251.                  cmd.Prepare()
1252.                  'Sets the parameters
1253.                  cmd.Parameters.AddWithValue("@PlayerID", IDNumber)
1254.                  'Executes sql query
1255.                  cmd.ExecuteNonQuery()
1256.                  'Closes database
1257.                  conn.Close()
1258.              Catch ex As Exception
1259.                  'Displays error message
1260.                  MsgBox(ex.ToString)
1261.              End Try
1262.          End Sub
1263.          Sub RemoveAllInsuranceFromDatabase(ByVal IDNumber As String)
1264.              Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1265.              Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1266.              conn.ConnectionString = sConnectionString
1267.              Try
1268.                  'Opens database
1269.                  conn.Open()
1270.                  cmd.Connection = conn
1271.                  'Removes insurance details for player who is to be removed
1272.                  cmd.CommandText = ("DELETE FROM `insurance` WHERE `ID_Number` =@IDNumber;")
1273.                  cmd.Prepare()
1274.                  'Sets the parameters
1275.                  cmd.Parameters.AddWithValue("@IDNumber", IDNumber)
1276.                  'Executes sql query
1277.                  cmd.ExecuteNonQuery()
1278.                  'Closes database
1279.                  conn.Close()
1280.              Catch ex As Exception
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1281.                 'Displays error message
1282.                 MsgBox(ex.ToString)
1283.             End Try
1284.         End Sub
1285.         Sub RemoveAllPlayerPositionPerferencesFromDatabase(ByVal IDNumber As String)
1286.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1287.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1288.             conn.ConnectionString = sConnectionString
1289.             Try
1290.                 'Opens database
1291.                 conn.Open()
1292.                 cmd.Connection = conn
1293.                 'Deletes players favourite position from database
1294.                 cmd.CommandText = ("DELETE FROM `players_preferred_position` WHERE `PlayerID` = @IDNumber;")
1295.                 cmd.Prepare()
1296.                 'Sets the parameters
1297.                 cmd.Parameters.AddWithValue("@IDNumber", IDNumber)
1298.                 'Executes command
1299.                 cmd.ExecuteNonQuery()
1300.                 'Closes connection to the database
1301.                 conn.Close()
1302.             Catch ex As Exception
1303.                 'Displays error messsage
1304.                 MsgBox(ex.ToString)
1305.             End Try
1306.         End Sub
1307.         Sub RemoveAllMVPPointsFromDatabase(ByVal IDNumber As String)
1308.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1309.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1310.             conn.ConnectionString = sConnectionString
1311.             Try
1312.                 'Opens database
1313.                 conn.Open()
1314.                 cmd.Connection = conn
1315.                 'Deletes players from player of the session
1316.                 cmd.CommandText = ("DELETE FROM `player_of_the_session` WHERE `PlayerID` = @PlayerID ;")
1317.                 cmd.Prepare()
1318.                 'Sets the parameters
1319.                 cmd.Parameters.AddWithValue("@PlayerID", IDNumber)
1320.                 'Executes command
1321.                 cmd.ExecuteNonQuery()
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1322.                 'Closes connection to the database
1323.                 conn.Close()
1324.             Catch ex As Exception
1325.                 MsgBox(ex.ToString)
1326.             End Try
1327.         End Sub
1328.         Sub RemoveAllPlayerOfTheSessionFromDatabase(ByVal IDNumber As String)
1329.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1330.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1331.             conn.ConnectionString = sConnectionString
1332.             Try
1333.                 'Opens database
1334.                 conn.Open()
1335.                 cmd.Connection = conn
1336.                 'Removes player of the session for the session
1337.                 cmd.CommandText = ("DELETE FROM `player_of_the_session` WHERE `PlayerID` = @PlayerID;")
1338.                 cmd.Prepare()
1339.                 'Sets the parameters
1340.                 cmd.Parameters.AddWithValue("@PlayerID", IDNumber)
1341.                 'Executes command
1342.                 cmd.ExecuteNonQuery()
1343.                 'Closes connection to the database
1344.                 conn.Close()
1345.             Catch ex As Exception
1346.                 MsgBox(ex.ToString)
1347.             End Try
1348.         End Sub
1349.         Function FindMVPPoints(ByVal PlayerID As Integer, ByVal RankingPosition As String, ByVal Season As String)
1350.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1351.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1352.             Dim dr As MySql.Data.MySqlClient.MySqlDataReader
1353.             Dim NumberOfTimesRead As String = 0
1354.             conn.ConnectionString = sConnectionString
1355.             Try
1356.                 'Opens connection to database
1357.                 conn.Open()
1358.                 cmd.Connection = conn
1359.                 'Reads number of mvp points the player has
1360.                 cmd.CommandText = ("SELECT Count(*) From `player_of_the_session`INNER Join `session` on `player_of_the_session`.`S
    essionID` = `session`.`SessionID` Where `player_of_the_session`.`Rank` = @Rank and `player_of_the_session`.`PlayerID` = @PlayerID and
     `session`.`Season` = @Season;")
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1361.                 cmd.Prepare()
1362.                 'Sets parameters value
1363.                 cmd.Parameters.AddWithValue("@Rank", RankingPosition)
1364.                 cmd.Parameters.AddWithValue("@PlayerID", PlayerID)
1365.                 cmd.Parameters.AddWithValue("@Season", Season)
1366.                 'Executes sql query
1367.                 dr = cmd.ExecuteReader()
1368.                 While dr.Read
1369.                     NumberOfTimesRead = dr("Count(*)")
1370.                 End While
1371.                 'Closes connection to database
1372.                 conn.Close()
1373.             Catch ex As Exception
1374.                 MsgBox(ex.ToString)
1375.             End Try
1376.             'Return value
1377.             Return NumberOfTimesRead
1378.         End Function
1379.         Sub FindEmailAddresses(ByRef PlayerList As ArrayList)
1380.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
1381.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
1382.             Dim dr As MySql.Data.MySqlClient.MySqlDataReader
1383.             conn.ConnectionString = sConnectionString
1384.             Try
1385.                 'Opens connection to database
1386.                 conn.Open()
1387.                 cmd.Connection = conn
1388.                 'Reads Email address
1389.                 cmd.CommandText = ("SELECT `Email Address` From `players`;")
1390.                 cmd.Prepare()
1391.                 dr = cmd.ExecuteReader()
1392.                 While dr.Read
1393.                     'Creates instance of player
1394.                     Dim Player As New Player(dr("Email Address"))
1395.                     'Adds Player to the list
1396.                     PlayerList.Add(Player)
1397.                 End While
1398.                 'Closes connection to the database
1399.                 conn.Close()
1400.             Catch ex As Exception
1401.                 'Displays error message
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                MsgBox(ex.ToString)
            End Try
        End Sub
    End Module
```

ADDS PLAYER FORM

```
Public Class AddPlayer
    Dim msSeason, msNewSeason, msYes, msNo As String
    Dim mbCanRegisterInsuranceForNextSeason As Boolean
    Dim mdDateToday As Date = Date.Today.AddMonths(5)
    Private Sub AddPlayer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        msYes = "Yes"
        msNo = "No"
        'Sets the maximum date you can pick as todays date
        DateOfBirthSelector.MaxDate = mdDateToday

        'Hides the items in the form about Netball Insurance until the user has entered their date of birth and they are an adult
        CurrentSeasonNetballInsuranceLBL.Hide()
        CurrentSeasonInsuranceSelector.Hide()
        CurrentSeasonInsuranceStartDateLBL.Hide()
        CurrentInsuranceStartDateSelector.Hide()

        NextSeasonInsuranceSelector.Hide()
        NextSeasonInsuranceLBL.Hide()
        NextSeasonInsuranceStartDate.Hide()
        NextSeasonInsuranceStartDateLBL.Hide()

        'Calculates the seasons for which insurance can be added
        CalculatesSeasonInsurance(mdDateToday, msSeason, msNewSeason, mbCanRegisterInsuranceForNextSeason)

        'Calculates and Sets Maximum and Minimum start date for Current Season Insurance
        CurrentInsuranceStartDateSelector.MinDate = mdDateToday
        CurrentInsuranceStartDateSelector.MaxDate = "31/08/" & Mid(msSeason, 4, 2)

        'If they can register insurance for next season
        If mbCanRegisterInsuranceForNextSeason Then
            'Calculates and Sets Maximum and Minimum start date for next Season Insurance
            NextSeasonInsuranceStartDate.MinDate = "01/09/" & Mid(msNewSeason, 1, 2)
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
33.             NextSeasonInsuranceStartDate.MaxDate = "31/08/" & Mid(msNewSeason, 4, 2)
34.         End If
35.
36.         'Modifies the labels for netball insurance selector with the calculated season
37.         CurrentSeasonNetballInsuranceLBL.Text = "Do you have Netball insurance for season " & msSeason & "?"
38.         NextSeasonInsuranceLBL.Text = "Do you have Netball insurance for season " & msNewSeason & "?"
39.
40.
41.
42.     End Sub
43.
44.     Private Sub BTN_Cancel_Click(sender As Object, e As EventArgs)
45.         Me.Close() 'If they cancel adding player sends back to main menu
46.     End Sub
47.
48.
49.     Private Sub DateOfBirthSelector_ValueChanged(sender As Object, e As EventArgs) Handles DateOfBirthSelector.ValueChanged
50.         'Calls function to check whether they are an adult
51.         If GetWhetherAdult(DateOfBirthSelector.Text, Date.Today) Then
52.
53.             'If they are an adult displays the details about the netball insurance and lets user selcet as they can only have netball
    insurance if they 18
54.             CurrentSeasonNetballInsuranceLBL.Show()
55.             CurrentSeasonInsuranceSelector.Show()
56.
57.
58.             'If in July and August display selector for next seasons insurance
59.             If mbCanRegisterInsuranceForNextSeason Then
60.                 NextSeasonInsuranceSelector.Show()
61.                 NextSeasonInsuranceLBL.Show()
62.             End If
63.         Else
64.             'If they are not older than 18 then removes the selection for whether they have insurance
65.             NextSeasonInsuranceSelector.Hide()
66.             NextSeasonInsuranceLBL.Hide()
67.             CurrentSeasonNetballInsuranceLBL.Hide()
68.             CurrentSeasonInsuranceSelector.Hide()
69.             CurrentInsuranceStartDateSelector.Hide()
70.             CurrentSeasonInsuranceStartDateLBL.Hide()
71.             NextSeasonInsuranceStartDate.Hide()
72.             NextSeasonInsuranceStartDateLBL.Hide()
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
        End If
    End Sub

    Private Sub BTN_AddPlayer_Click_1(sender As Object, e As EventArgs) Handles BTN_AddPlayer.Click
        Dim msFirstName, msSurname, msEmailAddress, msFavouritePosition, msSecondFavouritePosition, msThirdFavouritePosition, msCurrentSeasonInsuranceSelected, msNextSeasonInsuranceSelected As String
        Dim mbCanRegisterInsuranceForNextSeason, mbValidInsurancesInput, mBGenerateNumberSuccessfully, mBAddedPlayerSuccessfully, mBInsertedAttendanceIntoDatabase As Boolean
        Dim mdDOFB, mdStartDateInsuranceCurrentSeason, mdStartDateInsuranceNextSeason As Date
        Dim mIPlayerIDNumber As Integer

        'Sets variables to inputs
        msFavouritePosition = FavouritePositionSelector.Text
        msSecondFavouritePosition = SecondFavouritePositionSelector.Text
        msThirdFavouritePosition = ThirdFavouritePositionSelector.Text
        mdDOFB = DateOfBirthSelector.Text
        msCurrentSeasonInsuranceSelected = CurrentSeasonInsuranceSelector.Text
        msNextSeasonInsuranceSelected = NextSeasonInsuranceSelector.Text
        msFirstName = FirstNameTXTB.Text
        msSurname = SurnameTxtBox.Text
        mdStartDateInsuranceCurrentSeason = CurrentInsuranceStartDateSelector.Text
        mdStartDateInsuranceNextSeason = NextSeasonInsuranceStartDate.Text
        msEmailAddress = TXT_EmailAddress.Text

        'Checks whether player is an adult
        If GetWhetherAdult(mdDOFB, mdDateToday) = True Then

            'If they can register Insurance then sets variables to the input
            'Checks whether inputs for insurance are valid
                    If mbCanRegisterInsuranceForNextSeason Then
                        mbValidInsurancesInput = ValidateInsuranceInput(msCurrentSeasonInsuranceSelected, False) And ValidateInsuranceInput(msNextSeasonInsuranceSelected, False)
                    Else
                        msNextSeasonInsuranceSelected = msNo
                        mbValidInsurancesInput = ValidateInsuranceInput(msCurrentSeasonInsuranceSelected, False)
                    End If
                Else
                    msCurrentSeasonInsuranceSelected = msNo
                    msNextSeasonInsuranceSelected = msNo
                    mbValidInsurancesInput = True
                End If
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
111.
112.                 'Validates the inputs
113.                 If mbValidInsurancesInput And ValidNames(msFirstName, msSurname) And ValidatePositionsInput(msFavouritePosition, msSec
    ondFavouritePosition, msThirdFavouritePosition) And ValidateEmailAddress(msEmailAddress) Then
114.
115.                     'Generates IDNumber for the new player
116.                     mIPlayerIDNumber = ReadMaxIDNumber()
117.
118.
119.                     'Displays the new players ID Number to the user
120.                     MsgBox(msFirstName & " " & msSurname & " ID Number = " & mIPlayerIDNumber)
121.
122.                     'Adds player to the datbase
123.                     ADDPlayerToDatabase(mIPlayerIDNumber, msFirstName, msSurname, msFavouritePosition, msSecondFavouritePosition, msTh
    irdFavouritePosition, mdDOFB, msEmailAddress)
124.
125.
126.                     'If they have insurance then adds insurance details to the database
127.                     If msCurrentSeasonInsuranceSelected = msYes Then
128.                         AddInsuranceToDatabase(mIPlayerIDNumber, msSeason, mdStartDateInsuranceCurrentSeason)
129.                     End If
130.
131.                     'If they have insurance for next seasonthen adds insurance details to the database
132.                     If NextSeasonInsuranceSelector.Text = msYes Then
133.                         AddInsuranceToDatabase(mIPlayerIDNumber, msNewSeason, mdStartDateInsuranceNextSeason)
134.                     End If
135.
136.                     'Closes form
137.                     Me.Close()
138.
139.                 End If
140.
141.             End Sub
142.
143.             Private Sub BTN_Cancel_Click_1(sender As Object, e As EventArgs) Handles BTN_Cancel.Click
144.                 Me.Close()
145.             End Sub
146.             Private Sub NextSeasonInsuranceSelector_SelectedIndexChanged(sender As Object, e As EventArgs) Handles NextSeasonInsurance
    Selector.SelectedIndexChanged
147.                 'If they have selected they have insurance for the next season
148.                 If NextSeasonInsuranceSelector.Text = msYes Then
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
149.
150.                     'Displays the start date of insurance selector and label
151.                     NextSeasonInsuranceStartDate.Show()
152.                     NextSeasonInsuranceStartDateLBL.Show()
153.                 End If
154.             End Sub
155.
156.             Private Sub CurrentSeasonInsuranceSelector_SelectedIndexChanged(sender As Object, e As EventArgs) Handles CurrentSeasonIns
    uranceSelector.SelectedIndexChanged
157.                 'If they have selected they have insurance for the current season
158.                 If CurrentSeasonInsuranceSelector.Text = msYes Then
159.
160.                     'Displays the start date of insurance selector and label
161.                     CurrentInsuranceStartDateSelector.Show()
162.                     CurrentSeasonInsuranceStartDateLBL.Show()
163.
164.                 End If
165.             End Sub
166.         End Class
```

MODIFY PLAYER'S DETAILS

```
1.  Public Class ModifyDetails
2.      Dim mALPlayersList As New ArrayList
3.      Dim mbCanRegisterInsuranceForNextSeason, mbHasinsurcaneforcurrentseason, mbhasinsuranceforнextseason As Boolean
4.      Dim msSeason, msNextSeason, mSNo, mSYes As String
5.      Dim mdDateToday As Date = Date.Today.AddMonths(5)
6.      Private Sub ModifyDetails_Load(sender As Object, e As EventArgs) Handles MyBase.Load
7.          Dim msDisplayValue As String
8.          mSNo = "No"
9.          mSYes = "Yes"
10.
11.
12.         'Calculates the seasons that the player can select insurance for
13.         CalculatesSeasonInsurance(mdDateToday, msSeason, msNextSeason, mbCanRegisterInsuranceForNextSeason)
14.         'Reads list of players from the database
15.         ReadPlayersFromDatabase(mALPlayersList)
16.         'For each player reads whether has Netball insurance and adds whether the person has insurance
17.         For Count = 0 To mALPlayersList.Count - 1
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
            ReadWhetherHasInsurance(CInt(mALPlayersList(Count).GetIDNumber), msSeason)
            mALPlayersList(Count).AddCurrentSeasonInsurance(ReadWhetherHasInsurance(CInt(mALPlayersList(Count).GetIDNumber), msSeason
))
            'If its July or August checks whether the players has insurance for next season
            If mbCanRegisterInsuranceForNextSeason Then
                mALPlayersList(Count).AddNextSeasonInsurance(ReadWhetherHasInsurance(CInt(mALPlayersList(Count).GetIDNumber), msNextS
eason))
            End If
        Next


        'Hides the items in the form about Netball Insurance
        CurrentSeasonInsuranceLBL.Hide()
        CurrentSeasonInsuranceSelector.Hide()
        CurrentSeasonInsuranceStartDateLBL.Hide()
        CurrentInsuranceStartDateSelector.Hide()

        NextSeasonInsuranceSelector.Hide()
        NextSeasonInsuranceLBL.Hide()
        NextSeasonInsuranceStartDate.Hide()
        NextSeasonInsuranceStartDateLBL.Hide()

        'Calculates and Sets Maximum and Minimum start date for Current Season Insurance
        CurrentInsuranceStartDateSelector.MinDate = mdDateToday
        CurrentInsuranceStartDateSelector.MaxDate = "31/08/" & Mid(msSeason, 4, 2)
        If mdDateToday.Month = 7 Or mdDateToday.Month = 8 Then
            NextSeasonInsuranceStartDate.MinDate = "01/09/" & CInt(Mid(msNextSeason, 4, 2) - 1)
            NextSeasonInsuranceStartDate.MaxDate = "31/08/" & Mid(msNextSeason, 4, 2)
        End If


        'Adds each player to the player selector
        For Count = 0 To mALPlayersList.Count - 1
            msDisplayValue = mALPlayersList(Count).GetIDNumber & " - " & mALPlayersList(Count).GetFullName
            PlayerIDSelector.Items.Add(msDisplayValue)
        Next
        'Sets max date of birth to the date today
        DateOfBirthPicker.MaxDate = mdDateToday
        'Modifies the labels about the insurance with the correct seasons
        CurrentSeasonInsuranceLBL.Text = "Add Netball Insurance for season" & msSeason
        NextSeasonInsuranceLBL.Text = "Add Netball Insurance for season" & msNextSeason
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
57.     End Sub
58.     Private Sub BTN_Cancel_Click(sender As Object, e As EventArgs) Handles BTN_Cancel.Click
59.         'If they click to close form then close forms
60.         Me.Close()
61.     End Sub
62.     Private Sub BTN_ModifyPlayer_Click(sender As Object, e As EventArgs) Handles BTN_ModifyDetails.Click
63.         Dim msFirstName, msSurname, msFavouritePosition, msSecondFavouritePosition, msThirdFavouritePosition, msCurrentSeasonInsuranc
    eSelected, msNextSeasonInsuranceSelected, MsEmailAddress As String
64.         Dim mBCanRegisterInsuranceForNextSeason, mBValidInsurancesInput, mBValidPlayer, mBUpdateSuccesful, mBAddedNextSeasonInsurance
    Succesfully, mBAddedInsuranceSuccesfully As Boolean
65.         Dim mDDOFB, mdNextSeasonInsuranceStartDate, mdCurrentSeasonInsuranceStartDate As Date
66.         Dim mIPlayerID, playerindex As Integer
67.
68.         'Checks an exisiting player has been selected
69.         For Count = 0 To mALPlayersList.Count - 1
70.             If PlayerIDSelector.Text = mALPlayersList(Count).GetIDNumber & " - " & mALPlayersList(Count).GetFullName Then
71.                 mBValidPlayer = True
72.                 mIPlayerID = mALPlayersList(Count).GetIDNumber
73.                 playerindex = Count
74.             End If
75.         Next
76.
77.         'If a valid player has been selected
78.         If mBValidPlayer = True Then
79.
80.             'Sets variables to the inputted values
81.             msFavouritePosition = FavouritePositionSelector.Text
82.             msSecondFavouritePosition = SecondFavouritePositionSelector.Text
83.             msThirdFavouritePosition = ThirdFavouritePositionSelector.Text
84.             mDDOFB = DateOfBirthPicker.Text
85.             msCurrentSeasonInsuranceSelected = CurrentSeasonInsuranceSelector.Text
86.             msNextSeasonInsuranceSelected = NextSeasonInsuranceSelector.Text
87.             msFirstName = FirstNameTXTBox.Text
88.             msSurname = SurnameTXTBox.Text
89.             mdNextSeasonInsuranceStartDate = NextSeasonInsuranceStartDate.Text
90.             mdCurrentSeasonInsuranceStartDate = CurrentInsuranceStartDateSelector.Text
91.             MsEmailAddress = TXT_EmailAddress.Text
92.             'Checks whether the player is an adult
93.             If GetWhetherAdult(mDDOFB, mdDateToday) = True Then
94.
95.                 'If they can register Insurance then sets variables to the input
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                'Checks whether inputs for insurance are valid
                If mBCanRegisterInsuranceForNextSeason Then
                    mBValidInsurancesInput = (ValidateInsuranceInput(msCurrentSeasonInsuranceSelected, mALPlayersList(playerindex).IsInsured) And ValidateInsuranceInput(msNextSeasonInsuranceSelected, mALPlayersList(playerindex).IsInsuredForNextSeason))

                    Else
                        mBValidInsurancesInput = ValidateInsuranceInput(msCurrentSeasonInsuranceSelected, mALPlayersList(playerindex).IsInsured)
                    End If

                Else
                    msCurrentSeasonInsuranceSelected = mSNo
                    msNextSeasonInsuranceSelected = mSNo
                    mBValidInsurancesInput = True
                End If

                'Validates the inputs for the players name and preferred positions
                If mBValidInsurancesInput And ValidNames(msFirstName, msSurname) And ValidatePositionsInput(msFavouritePosition, msSecondFavouritePosition, msThirdFavouritePosition) And ValidateEmailAddress(MsEmailAddress) Then

                    'Modifies the players details in the database
                    UpdateDatabase(mIPlayerID, msFirstName, msSurname, msFavouritePosition, msSecondFavouritePosition, msThirdFavouritePosition, mDDOFB, MsEmailAddress)
                    UpdatePlayersPosition(msFavouritePosition, mIPlayerID, 1)
                    UpdatePlayersPosition(msSecondFavouritePosition, mIPlayerID, 2)
                    UpdatePlayersPosition(msThirdFavouritePosition, mIPlayerID, 3)
                    'If the player has insurance
                    If CurrentSeasonInsuranceSelector.Text = mSYes Then

                        'Adds the players insurance details to the database
                        AddInsuranceToDatabase(mIPlayerID, msSeason, mdCurrentSeasonInsuranceStartDate)
                    Else
                        mBAddedInsuranceSuccesfully = True
                    End If

                    'If player has insurance for next season
                    If NextSeasonInsuranceSelector.Text = mSYes Then

                        'Adds the players insurance details to the database
                        AddInsuranceToDatabase(mIPlayerID, msNextSeason, mdNextSeasonInsuranceStartDate)
                    Else
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                            mBAddedNextSeasonInsuranceSuccesfully = True
                        End If
                    End If
                    'Displays message
                    MsgBox("Added succesfully")
                    'Closes database
                    Me.Close()
                Else
                    'Displays error message
                    MsgBox("Invalid Player selected")
                End If
            End Sub
            Private Sub CurrentSeasonInsuranceSelector_SelectedIndexChanged(sender As Object, e As EventArgs) Handles CurrentSeasonInsuranceSelector.SelectedIndexChanged
                'If they have selected they have insurance then displays label and date selector for user to select start date of insurance
                If CurrentSeasonInsuranceSelector.Text = mSYes Then
                    CurrentInsuranceStartDateSelector.Show()
                    CurrentSeasonInsuranceStartDateLBL.Show()
                End If
            End Sub

            Private Sub NextSeasonInsuranceSelector_SelectedIndexChanged(sender As Object, e As EventArgs) Handles NextSeasonInsuranceSelector.SelectedIndexChanged
                'If they have selected they have insurance then displays label and date selector for user to select start date of insurance
                If NextSeasonInsuranceSelector.Text = mSYes Then
                    NextSeasonInsuranceStartDate.Show()
                    NextSeasonInsuranceStartDateLBL.Show()
                End If
            End Sub

            Private Sub DateOfBirthPicker_ValueChanged(sender As Object, e As EventArgs) Handles DateOfBirthPicker.ValueChanged
                'Checks whether player is adult
                If GetWhetherAdult(DateOfBirthPicker.Text, mdDateToday) And mbHasinsurcaneforcurrentseason = False Then

                    'If they are an adult displays the details about the netball insurance and lets user selcet as they can only have netball insurance if they 18
                    CurrentSeasonInsuranceLBL.Show()
                    CurrentSeasonInsuranceSelector.Show()

```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
169.                 'If the user can added insurance for next season displays the items allowing user to select whether they have insu
    rance for next season
170.                 If mdDateToday.Month = 7 Or mdDateToday.Month = 8 And mbhasinsurancefornextseason = False Then
171.                     NextSeasonInsuranceSelector.Show()
172.                     NextSeasonInsuranceLBL.Show()
173.
174.                 End If
175.             Else
176.
177.                 'If they are not older than 18 then removes the selection for whether they have insurance
178.                 CurrentSeasonInsuranceLBL.Hide()
179.                 CurrentSeasonInsuranceSelector.Hide()
180.             End If
181.         End Sub
182.
183.         Private Sub PlayerIDSelector_SelectedIndexChanged(sender As Object, e As EventArgs) Handles PlayerIDSelector.SelectedIndex
    Changed
184.             'When a differnet player has been selected
185.             Dim mSIndexOfSelectedPlayer As String = ""
186.             Dim mBValidPlayer As Boolean
187.
188.             'Iterates through list of players
189.             For Count = 0 To mALPlayersList.Count - 1
190.
191.                 'If selected player matches player in the list
192.                 If PlayerIDSelector.Text = mALPlayersList(Count).GetIDNumber & " - " & mALPlayersList(Count).GetFullName Then
193.
194.                     'Set to the index of the player who is selected
195.                     mSIndexOfSelectedPlayer = Count
196.
197.                     'Sets each textbox/date selector/position selector to the current players value
198.                     FirstNameTXTBox.Text = mALPlayersList(mSIndexOfSelectedPlayer).GetFirstName
199.                     SurnameTXTBox.Text = mALPlayersList(mSIndexOfSelectedPlayer).GetSurname
200.                     FavouritePositionSelector.Text = mALPlayersList(mSIndexOfSelectedPlayer).GetFavouritePosition
201.                     SecondFavouritePositionSelector.Text = mALPlayersList(mSIndexOfSelectedPlayer).GetSecondFavouritePosition
202.                     ThirdFavouritePositionSelector.Text = mALPlayersList(mSIndexOfSelectedPlayer).GetThirdFavouritePosition
203.                     DateOfBirthPicker.Text = mALPlayersList(mSIndexOfSelectedPlayer).GetDateOfBirth
204.                     TXT_EmailAddress.Text = mALPlayersList(mSIndexOfSelectedPlayer).GetEmailAddress
205.
206.                     'Valid player has been selected
207.                     mBValidPlayer = True
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
208.                     End If
209.                 Next
210.
211.                 'If valid player has been selected
212.                 If mBValidPlayer = True Then
213.
214.                     'Checks whether selected player is an adult
215.                     If GetWhetherAdult(mALPlayersList(mSIndexOfSelectedPlayer).GetDateOfBirth, mdDateToday) Then
216.
217.                         MsgBox(mALPlayersList(mSIndexOfSelectedPlayer).IsInsured)
218.
219.                         'If player has insurance hides the labels and selector for the current season of insurance to the user
220.                         If mALPlayersList(mSIndexOfSelectedPlayer).IsInsured = True Then
221.
222.                             mbHasinsurcaneforcurrentseason = True
223.                             CurrentSeasonInsuranceSelector.Hide()
224.                             CurrentSeasonInsuranceLBL.Hide()
225.                             'If they don't have insurance then displays the label and selector for the user whether they have insuranc
    e
226.                         Else
227.                             CurrentSeasonInsuranceSelector.Show()
228.                             CurrentSeasonInsuranceLBL.Show()
229.                         End If
230.
231.                         'If its July or August
232.                         If mbCanRegisterInsuranceForNextSeason Then
233.
234.                             'If player has insurance hides the labels and selector for next season of insurance to the user
235.                             If mALPlayersList(mSIndexOfSelectedPlayer).IsInsuredForNextSeason = True Then
236.                                 mbhasinsuranceforнextseason = True
237.                                 NextSeasonInsuranceSelector.Hide()
238.                                 NextSeasonInsuranceLBL.Hide()
239.
240.                                 'If they don't have insurance then displays the label and selector for the user whether they have insu
    rance
241.                             ElseIf mALPlayersList(mSIndexOfSelectedPlayer).IsInsured = True Then
242.                                 NextSeasonInsuranceSelector.Show()
243.                                 NextSeasonInsuranceLBL.Show()
244.                             End If
245.                         End If
246.                         'If they aren't an adult then hide label and selector for whether the user has insurance
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                    Else
                        CurrentSeasonInsuranceSelector.Hide()
                        CurrentSeasonInsuranceLBL.Hide()
                        NextSeasonInsuranceLBL.Hide()
                        NextSeasonInsuranceSelector.Hide()
                    End If
                End If
            End Sub
        End Class
```

REMOVE PLAYER FORM

```
Public Class RemovePlayer
    Dim PlayersList As New ArrayList
    Private Sub RemovePlayer_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Dim mSDisplayValue As String

        'Reads the players from the database
        ReadPlayersFromDatabase(PlayersList)

        'Displays and add each player to the list of players to select
        For Count = 0 To PlayersList.Count - 1
            mSDisplayValue = PlayersList(Count).GetIDNumber & " - " & PlayersList(Count).GetFullName
            PlayerIDSelector.Items.Add(mSDisplayValue)
        Next


    End Sub

    Private Sub BTN_Cancel_Click(sender As Object, e As EventArgs) Handles BTN_Cancel.Click
        'Cancel button pressed and then form closed
        Me.Close()
    End Sub

    Private Sub BTN_RemovePlayer_Click(sender As Object, e As EventArgs) Handles BTN_RemovePlayer.Click
        Dim mSValidPlayer As Boolean
        Dim mSPlayerToBeRemoved As Integer
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
26.
27.         'Checks player selected is a exisiting player
28.         For Count = 0 To PlayersList.Count - 1
29.             If PlayerIDSelector.Text = PlayersList(Count).GetIDNumber & " - " & PlayersList(Count).GetFullName Then
30.                 mSPlayerToBeRemoved = PlayersList(Count).GetIDNumber
31.                 mSValidPlayer = True
32.             End If
33.         Next
34.
35.         'If valid player selected
36.         If mSValidPlayer = True Then
37.
38.             'Removes player from database
39.             DeletePlayerFromDatabase(mSPlayerToBeRemoved)
40.             RemoveAllAttendancesFromDatabase(mSPlayerToBeRemoved)
41.             RemoveAllInsuranceFromDatabase(mSPlayerToBeRemoved)
42.             RemoveAllPlayerPositionPerferencesFromDatabase(mSPlayerToBeRemoved)
43.             RemoveAllMVPPointsFromDatabase(mSPlayerToBeRemoved)
44.             'If successfully removed player then then closes form and clears list
45.             PlayersList.Clear()
46.             Me.Close()
47.
48.             'Displays error message
49.         Else
50.             MsgBox("Invalid player selected")
51.         End If
52.     End Sub
53. End Class
```

ADD SESSION FORM

```
1.  Imports System.Net.Mail
2.  Public Class AddSession
3.      Dim mDDateToday As Date
4.      Dim PlayerList As New ArrayList
5.      Dim mSStartTime, msEndTime, msLocation, msSeason, msNewSeason As String
6.      Dim mDDate As Date
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
7.      Private Sub AddSession_Load(sender As Object, e As EventArgs) Handles MyBase.Load
8.          'Loads the page
9.          'Adds minute and hour for the session start time and end time
10.         StartMinuteSelector.Items.Add("00")
11.         EndMinuteSelector.Items.Add("00")
12.         StartMinuteSelector.Items.Add("05")
13.         EndMinuteSelector.Items.Add("05")
14.         For Count = 10 To 59 Step 5
15.             StartMinuteSelector.Items.Add(Count)
16.             EndMinuteSelector.Items.Add(Count)
17.         Next
18.         For Count = 0 To 9
19.             EndHourSelector.Items.Add("0" & Count)
20.             StartHourSelector.Items.Add("0" & Count)
21.         Next
22.         For Count = 10 To 23
23.             EndHourSelector.Items.Add(Count)
24.             StartHourSelector.Items.Add(Count)
25.         Next
26.         'Sets date today
27.         mDDateToday = Date.Today
28.         'Sets minimum date for session to date today
29.         DateSelector.MinDate = mDDateToday
30.         BTN_Reminder.Hide()
31.     End Sub
32.     Private Sub BTN_AddSession_Click(sender As Object, e As EventArgs) Handles BTN_AddSession.Click
33.         Dim mISessionIDNumber As Integer
34.         Dim mbCanRegisterInsuranceForNextSeason As Boolean
35.         Dim mbValidInputs As Boolean
36.         'Validates the sessions index
37.         Try
38.             'Checks start time is before end time
39.             If CInt(StartHourSelector.Text) > CInt(EndHourSelector.Text) Or (CInt(StartHourSelector.Text) = CInt(EndHourSelector.Text
    ) And CInt(EndMinuteSelector.Text) <= CInt(StartMinuteSelector.Text)) Or TXTBox_Location.Text = "" Or CInt(StartMinuteSelector.Text)
    > 59 Or CInt(StartMinuteSelector.Text) < 0 Or CInt(EndMinuteSelector.Text) > 59 Or CInt(EndMinuteSelector.Text) < 0 Or CInt(StartHour
    Selector.Text) > 24 Or CInt(StartHourSelector.Text) < 0 Or CInt(EndHourSelector.Text) > 24 Or CInt(EndHourSelector.Text) < 0 Then
40.                 MsgBox("Invalid input")
41.                 mbValidInputs = False
42.             Else
43.                 'Gets session details which have been entered
44.                 mSStartTime = StartHourSelector.Text & ":" & StartMinuteSelector.Text
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                msEndTime = EndHourSelector.Text & ":" & EndMinuteSelector.Text
                mDDate = DateSelector.Text
                msLocation = TXTBox_Location.Text
                mbValidInputs = True
            End If
        Catch ex As Exception
            mbValidInputs = False
            'Displays error message
            MsgBox("Error with inputs")
        End Try
        'Adds session to the database if valid input
        If mbValidInputs Then
            'Finds session id number
            ReadIDNumber(mISessionIDNumber)
            If mISessionIDNumber <> 0 Then
                'Calculates season for the session
                CalculatesSeasonInsurance(mDDate, msSeason, msNewSeason, mbCanRegisterInsuranceForNextSeason)
                'Adds session to the database
                AddSessionToDatabase(mSStartTime, msEndTime, msSeason, mDDate, msLocation, mISessionIDNumber)
                BTN_AddSession.Hide()
                BTN_Reminder.Show()
            Else
                'Displays error message
                MsgBox("Failed to generate id number")
            End If
        End If
    End Sub
    Private Sub BTN_RemovePlayer_Click(sender As Object, e As EventArgs) Handles BTN_CancelSession.Click
        Me.Close()
    End Sub
    Private Sub BTN_Reminder_Click(sender As Object, e As EventArgs) Handles BTN_Reminder.Click
        FindEmailAddresses(PlayerList)
        Dim EmailBody As String
        'Creates the email body
        EmailBody = "Hi All,"
        EmailBody = EmailBody & Environment.NewLine & "A new session has been added online"
        EmailBody = EmailBody & Environment.NewLine & "Please can update whether you are available to play by following the link belo
w"
        EmailBody = EmailBody & Environment.NewLine & "Session Date - " & mDDate
        EmailBody = EmailBody & Environment.NewLine & "Start Time - " & mSStartTime
        EmailBody = EmailBody & Environment.NewLine & "End Time - " & msEndTime
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
        EmailBody = EmailBody & Environment.NewLine & "Location - " & msLocation
        EmailBody = EmailBody & Environment.NewLine & "http://localhost:60590/WebForm1"
        EmailBody = EmailBody & Environment.NewLine & "Thanks,"
        EmailBody = EmailBody & Environment.NewLine & "Rodborough Rockets Manager"
        Try
            Dim Smtp_Server As New SmtpClient
            Dim e_mail As New MailMessage()
            Smtp_Server.UseDefaultCredentials = False
            Smtp_Server.Credentials = New Net.NetworkCredential("rodboroughrocketclub@gmail.com", "CompSciNea2020")
            Smtp_Server.Port = 587
            Smtp_Server.EnableSsl = True
            Smtp_Server.Host = "smtp.gmail.com"
            e_mail = New MailMessage()
            e_mail.From = New MailAddress("rodboroughrocketclub@gmail.com")
            'Gets email addresses
                    For Count = 0 To PlayerList.Count - 1
                        e_mail.Bcc.Add(PlayerList(Count).GetEmailAddress())
                    Next
                    'Sets subject
                    e_mail.Subject = "New Rodborough Rockets Session"
                    e_mail.IsBodyHtml = False
                    e_mail.Body = EmailBody
                    'Sends email
                    Smtp_Server.Send(e_mail)
                    MsgBox("Mail Sent")
                Catch error_t As Exception
                    MsgBox(error_t.ToString)
                End Try
                Me.Close()
            End Sub
        End Class
```

ADD ATTENDANCE

```
Public Class AddAttendance
    Dim MsDateToday As Date = Date.Today
    Dim PlayersList As New ArrayList
    Dim SessionList As New ArrayList
    Private Sub AddAttendance_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
        Dim mSDisplayValue As String
        'Reads the list of players from the database
        ReadPlayersFromDatabase(PlayersList)
        For Count = 0 To PlayersList.Count - 1
            mSDisplayValue = PlayersList(Count).GetIDNumber & " - " & PlayersList(Count).GetFullName
            PlayerIDSelector.Items.Add(mSDisplayValue)
        Next

        'Reads the list of sessions from the database
        ReadSessionsDetailsFromDatabase(SessionList, MsDateToday)
        For Count = 0 To SessionList.Count - 1
            SelectSession.Items.Add(SessionList(Count).GetDisplayValue)
        Next

    End Sub

    Private Sub BTN_AddAttendance_Click(sender As Object, e As EventArgs) Handles BTN_AddAttendance.Click
        Dim mbValidSession, mbValidPlayer, mbValidInput, mBHasAlreadyGivenAvailability As Boolean
        Dim msYes, msNo, mSPlayerIDNumber, mSSessionID, mSDateOfSession As String
        Dim DofB As Date
        'Validates the users inputs
        msYes = "Yes"
        msNo = "No"
        'Validates player selected
        For Count = 0 To PlayersList.Count - 1
            If PlayerIDSelector.Text = PlayersList(Count).GetIDNumber & " - " & PlayersList(Count).GetFullName Then
                mbValidPlayer = True
                mSPlayerIDNumber = PlayersList(Count).GetIDNumber
                DofB = PlayersList(Count).GetDateOfBirth
            End If
        Next
        'Validates session selected
        For Count = 0 To SessionList.Count - 1
            If SelectSession.Text = SessionList(Count).GetDisplayValue Then
                mbValidSession = True
                mSDateOfSession = SessionList(Count).GetSessionDate
                mSSessionID = SessionList(Count).GetIDNumber
            End If
        Next
        If AvailableForSession.Text = msYes Or AvailableForSession.Text = msNo Then
            mbValidInput = True
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
47.         End If
48.         'Adds the players attendance to the database
49.         If mbValidInput And mbValidSession And mbValidPlayer Then
50.             'Checks whether player had already given attendance
51.             mBHasAlreadyGivenAvailability = ReadWhetherGivenAvailability(mSPlayerIDNumber, mSSessionID)
52.             'Removes attendance if they can't attend and if they have said whether they can attend previously
53.             If mBHasAlreadyGivenAvailability = True And AvailableForSession.Text = msNo Then
54.                 RemoveAttendance(mSSessionID, mSPlayerIDNumber)
55.             End If
56.             'Adds attendance if they can attend
57.             If mBHasAlreadyGivenAvailability = False And AvailableForSession.Text = msYes Then
58.                 AddAttendanceToDatabase(mSSessionID, mSPlayerIDNumber)
59.                 'Calculates amount due for the session
60.                 MsgBox("Amount due - " & CalculateAmountDue(mSPlayerIDNumber, mSDateOfSession, DofB))
61.             End If
62.             Me.Close()
63. 
64.                 'Displays error messages
65.             ElseIf mbValidInput = False Then
66.             MsgBox("Please select whether you are available")
67.         ElseIf mbValidSession = False Then
68.             MsgBox("Please select a valid session")
69.         ElseIf mbValidPlayer = False Then
70.             MsgBox("Please select a valid player")
71.         End If
72.     End Sub
73. 
74.     Private Sub BTN_Cancel_Click(sender As Object, e As EventArgs) Handles BTN_Cancel.Click
75.         Me.Close()
76.     End Sub
77. End Class
78. Public Class Sessions
79.     Protected IDNumber, Location, StartTime, EndTime, Season As String
80.     Protected DateOfSession As Date
81.     Public Sub New(ByVal IDNumber As String, ByVal Location As String, ByVal DateOfSession As Date, ByVal StartTime As String, ByVal
    EndTime As String)
82.         Me.IDNumber = IDNumber
83.         Me.Location = Location
84.         Me.DateOfSession = DateOfSession
85.         Me.StartTime = StartTime
86.         Me.EndTime = EndTime
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
87.     End Sub
88.     Public Sub New(ByVal IDNumber As String, ByVal Location As String, ByVal DateOfSession As Date, ByVal StartTime As String)
89.         Me.IDNumber = IDNumber
90.         Me.Location = Location
91.         Me.DateOfSession = DateOfSession
92.         Me.StartTime = StartTime
93.     End Sub
94.
95.     Public Sub New(ByVal IDNumber As String, ByVal Location As String, ByVal Season As String, ByVal DateOfSession As String, ByVal S
    tartTime As String, ByVal RandomChance As Integer)
96.         Me.IDNumber = IDNumber
97.         Me.Location = Location
98.         Me.DateOfSession = DateOfSession
99.         Me.StartTime = StartTime
100.                Me.Season = Season
101.            End Sub
102.            'Returns display value
103.            Function GetDisplayValue() As String
104.                Dim msValue
105.                msValue = DateOfSession.ToShortDateString & " - " & Mid(StartTime, 1, 5) & " @ " & Location
106.                Return msValue
107.            End Function
108.            'Returns id number
109.            Function GetIDNumber() As String
110.                Return IDNumber
111.            End Function
112.            'Returns session date
113.            Function GetSessionDate() As Date
114.                Return DateOfSession
115.            End Function
116.            'Returns season
117.            Function GetSeason() As String
118.                Return Season
119.            End Function
120.        End Class
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

AWARD MVP POINTS FORM

```
1.  Public Class AwardMVPPoints
2.      Dim MALSessionList As New ArrayList
3.      Dim MALPlayersList As New ArrayList
4.      Dim mdDateToday As Date = Date.Today
5.      Dim miIndex As Integer
6.      Private Sub AwardMVPPoints_Load(sender As Object, e As EventArgs) Handles MyBase.Load
7.          'Reads the sessions
8.          ReadListOfSessions(MALSessionList, mdDateToday)
9.
10.         'Adds list of sessions to the listbpx
11.         For Count = 0 To MALSessionList.Count - 1
12.             SessionSelection.Items.Add(MALSessionList(Count).GetDisplayValue)
13.         Next
14.
15.     End Sub
16.
17.     Private Sub SessionSelection_SelectedIndexChanged(sender As Object, e As EventArgs) Handles SessionSelection.SelectedIndexChanged

18.         'Clears list of player
19.         MALPlayersList.Clear()
20.         'Cleared items when session read
21.         BestPlayerSelector.Items.Clear()
22.         SecondBestPlayerSelector.Items.Clear()
23.         ThirdBestPlayerSelector.Items.Clear()
24.         'Gets index of session selected
25.         For Count = 0 To MALSessionList.Count - 1
26.             If SessionSelection.Text = MALSessionList(Count).GetDisplayValue Then
27.                 miIndex = Count
28.             End If
29.         Next
30.         'Reads the list of players who attended the session
31.         ReadListOfPlayers(MALPlayersList, MALSessionList(miIndex).GetIDNumber)
32.         'Adds the players to the list box for best, second and third player of the session list box
33.         For Count = 0 To MALPlayersList.Count - 1
34.             BestPlayerSelector.Items.Add(MALPlayersList(Count).GetFullName)
35.             SecondBestPlayerSelector.Items.Add(MALPlayersList(Count).GetFullName)
36.             ThirdBestPlayerSelector.Items.Add(MALPlayersList(Count).GetFullName)
37.         Next
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
38.     End Sub
39.
40.     Private Sub BTN_SubmitMVPPoints_Click(sender As Object, e As EventArgs) Handles BTN_SubmitMVPPoints.Click
41.         Dim msBestPlayerID, msSecondBestPlayerID, msThirdBestPlayerID As String
42.         Dim ValidPlayer1, ValidPlayer2, ValidPlayer3 As Boolean
43.         msBestPlayerID = -1
44.         msSecondBestPlayerID = -1
45.         msThirdBestPlayerID = -1
46.         'Validates the users inputs
47.         ValidPlayer1 = False
48.         ValidPlayer2 = False
49.         ValidPlayer3 = False
50.         For Count = 0 To MALPlayersList.Count - 1
51.             'Checks if player has been selected for best players of the session
52.             If MALPlayersList(Count).GetFullName = BestPlayerSelector.Text Then
53.                 msBestPlayerID = MALPlayersList(Count).GetIDNumber
54.                 ValidPlayer1 = True
55.             End If
56.             'Checks if player has been selected for second best players of the session
57.             If MALPlayersList(Count).GetFullName = SecondBestPlayerSelector.Text Then
58.                 msSecondBestPlayerID = MALPlayersList(Count).GetIDNumber
59.                 ValidPlayer2 = True
60.             End If
61.             'Checks if player has been selected for third best players of the session
62.             If MALPlayersList(Count).GetFullName = ThirdBestPlayerSelector.Text Then
63.                 msThirdBestPlayerID = MALPlayersList(Count).GetIDNumber
64.                 ValidPlayer3 = True
65.             End If
66.         Next
67.         'Invalid inputs and then display error message
68.         If ValidPlayer1 = False Then
69.             MsgBox("Please select a valid player for best player")
70.         End If
71.         If ValidPlayer2 = False Then
72.             MsgBox("Please select a valid player for 2rd best player")
73.         End If
74.         If ValidPlayer3 = False Then
75.             MsgBox("Please select a valid player for 3rd best player")
76.         End If
77.         'Checks if the same player has been selected multiple times
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
        If msThirdBestPlayerID = msSecondBestPlayerID Or msThirdBestPlayerID = msBestPlayerID Or msBestPlayerID = msSecondBestPlayerI
    D Then
            MsgBox("invalid input")
        Else
            If ValidPlayer1 = True And ValidPlayer2 = True And ValidPlayer3 = True Then
                'Adds the points to the database
                AddMVPPointsToDatabase(msBestPlayerID, MALSessionList(miIndex).GetIDNumber, 1)
                AddMVPPointsToDatabase(msSecondBestPlayerID, MALSessionList(miIndex).GetIDNumber, 2)
                AddMVPPointsToDatabase(msThirdBestPlayerID, MALSessionList(miIndex).GetIDNumber, 3)
                'Closes form
                Me.Close()
            End If
        End If
    End Sub
    'Closes form
    Private Sub BTN_Cancel_Click(sender As Object, e As EventArgs) Handles BTN_Cancel.Click
        Me.Close()
    End Sub
End Class
```

GENERATE TEAMS

```
Imports System.Net.Mail
Public Class GenerateTeams
    Dim ReplacementPlayerAtTheMoment(6, 2) As String
    Dim ReplacementPlayerTwoAtTheMoment(6, 2) As String
    Dim ReplacementPlayerThreeAtTheMoment(6, 2) As String
    Dim MALPlayerList As New ArrayList
    Dim MALSessionsList As New ArrayList
    Dim TeamA As New Team
    Dim TeamB As New Team
    Dim IndexValue As String
    Dim mdDateToday As Date = Date.Today
    Dim Positions() As String = {"GK", "GD", "WD", "C", "WA", "GA", "GS"}
    Dim NumberOfPlayersNeededPerPosition() = {2, 2, 2, 2, 2, 2, 2}
    Dim NumberOfPlayersPerPosition(7, 7) As String
    Dim NumberOfPositionsWithCorrectAmount As Integer
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
    Dim NumberOfPositionsWithTooLittle As Integer
    Dim NumberOfPositionsWithTooMany As Integer
    Dim FloatingCentre As Integer = -1
    Private Sub GenerateTeams_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        'Loads the form and hides image of court and hides teams label
        BTN_SaveTeam.Hide()
        BTN_SendTeam.Hide()
        LBL_TeamA.Hide()
        LBL_TeamB.Hide()
        LBLTeamAPlayers.Hide()
        LBLTeamBPlayers.Hide()
        LBLFloatingCentre.Hide()
        P9.Hide()
        P10.Hide()
        P11.Hide()
        P12.Hide()
        P13.Hide()
        P14.Hide()
        AStats.Hide()
        BStats.Hide()
        'Reads the list of session
        ReadListOfSessionsToGenerateTeams(mdDateToday, MALSessionsList)
        'Adds each session to the drop down box
        For Count = 0 To MALSessionsList.Count - 1
            SessionSelection.Items.Add(MALSessionsList(Count).GetDisplayValue)
        Next
    End Sub
    Private Sub BTN_GenerateTeams_Click(sender As Object, e As EventArgs) Handles BTN_GenerateTeams.Click
        Dim msselectedSession As String
        Dim miSelectedSessionIndex As Integer
        Dim msPlayerID As String
        Dim mbValidSession As Boolean
        msselectedSession = SessionSelection.Text


        'Checks a valid session has been selected and stores the index of the session
        For Count = 0 To MALSessionsList.Count - 1
            If SessionSelection.Text = MALSessionsList(Count).GetDisplayValue Then
                miSelectedSessionIndex = MALSessionsList(Count).GetIDNumber
                mbValidSession = True
                IndexValue = Count
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
            End If
        Next

        'Reads the list of players for the session
        If mbValidSession Then
            ReadListOfPlayers(MALPlayerList, miSelectedSessionIndex)

            'Gets Season of the session
            Dim msSeason As String
            msSeason = MALSessionsList(IndexValue).GetSeason

            'Finds the number of MVP Points for each player
            For Count = 0 To MALPlayerList.Count - 1
                msPlayerID = MALPlayerList(Count).GetIDNumber
                For index = 1 To 3
                    MALPlayerList(Count).AddMVPPoint(FindMVPPoints(msPlayerID, index, msSeason) * (4 - index))
                Next
            Next


            'Sets the number of players needed for each player depending on the number of players
            'When 14 or more players 2 players needed for each position
            If MALPlayerList.Count >= 14 Then
                'Stays the same
                'When 12 players, 2 players needed for each position apart from WD where 0 players are needed
            ElseIf MALPlayerList.Count = 12 Then
                NumberOfPlayersNeededPerPosition(2) = 0
                'When 13 players, 2 players needed for each position apart from C where 1 player are needed
            ElseIf MALPlayerList.Count = 13 Then
                NumberOfPlayersNeededPerPosition(3) = 1
                'When 11 players, 2 players needed for each position apart from WD where 0 players are needed and C where 1 player is
    needed
            ElseIf MALPlayerList.Count = 11 Then
                NumberOfPlayersNeededPerPosition(3) = 1
                NumberOfPlayersNeededPerPosition(2) = 0
                'When 10 players, 2 players needed for each position apart from WD where 0 players are needed and WA where 0 player i
   s needed
            ElseIf MALPlayerList.Count = 10 Then
                NumberOfPlayersNeededPerPosition(2) = 0
                NumberOfPlayersNeededPerPosition(4) = 0
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                'When 9 players, 2 players needed for each position apart from WD where 0 players are needed and WA where 0 player is needed and C where 1 player is needed
            ElseIf MALPlayerList.Count = 9 Then
                NumberOfPlayersNeededPerPosition(3) = 1
                NumberOfPlayersNeededPerPosition(2) = 0
                NumberOfPlayersNeededPerPosition(4) = 0
                End If

                'Checks to see if there is the correct number of players who preferred position is that position for each position needed
                FindNumberOfPlayerForEachPosition()
                'There is too little players for at least one position
                If NumberOfPositionsWithTooLittle > 0 Then
                    'Finds player who can play each position if there isn't the correct number of players for position
                    FindReplacementPlayersSecondrayPosition()
                    'Checks whether there is the correct number of players for each positon
                    If FitsSquad() = False Then
                        'If it doesn't fit team then finds players who third favourite position is position which need player
                        FindReplacementPlayersThirdPosition()
                        'Checks whether there is the correct number of players for each positon
                        If FitsSquad() = False Then
                            'If it doesn't fit team then finds players based on area of their favourite position is same area as position which needs players
                            FindAreaReplacement()
                            'Checks whether there is the correct number of players for each positon
                            If FitsSquad() = False Then
                                'If it doesn't fit team then finds players based on area of their second favourite position is same area as position which needs players
                                FindArea2Replacement()
                                'Checks whether there is the correct number of players for each positon
                                If FitsSquad() = False Then
                                    'If it doesn't fit team then finds players based on area of their third favourite position is same area as position which needs players
                                    FindArea3Replacement()
                                    'Checks whether there is the correct number of players for each positon
                                    If FitsSquad() = False Then
                                        'Finds random player for position which needs player
                                        FindRandomPlayer()
                                    End If
                                End If
                            End If
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
131.                         End If
132.                     End If
133.                 End If
134.
135.                 'Assigns a position to each player if they haven't already been assigned a position
136.                 AssignPositions()
137.                 'Assigns the player to a team depending on their postion
138.                 AssignPlayersToATeam()
139.                 'Assigns the subs to a team
140.                 AssignSubsToATeam()
141.                 'Displays the team in the form
142.                 DisplayTeams()
143.             Else
144.                 'Displays error message
145.                 MsgBox("Please enter a valid session")
146.             End If
147.         End Sub
148.         Sub AssignSubsToATeam()
149.             Dim NumberOfSubs As Integer = 0
150.             Dim Subs(MALPlayerList.Count) As String
151.             'Loops through every player and finds who is a sub and if they are then saves there index number
152.             For Count = 0 To MALPlayerList.Count - 1
153.                 If MALPlayerList(Count).GetPosition = "SUB" Then
154.                     NumberOfSubs = NumberOfSubs + 1
155.                     'Stores index of sub in the list
156.                     Subs(NumberOfSubs) = Count
157.                 End If
158.             Next
159.             'Loops through sub and allocates them to a team depending on the number of subs for each team
160.             For Count = 1 To NumberOfSubs
161.                 'If Team A has more subs then joins TeamB
162.                 If TeamA.NumberOfSums > TeamB.NumberOfSums Then
163.                     TeamB.AddPlayerToSub(MALPlayerList(Subs(Count)), (MALPlayerList(Subs(Count)).GetNumberOfMVPPoints / 5))
164.                     'If Team B has more subs then joins TeamA
165.                 ElseIf TeamB.NumberOfSums > TeamA.NumberOfSums Then
166.                     TeamA.AddPlayerToSub(MALPlayerList(Subs(Count)), (MALPlayerList(Subs(Count)).GetNumberOfMVPPoints / 5))
167.                     'They have equal sub
168.                 Else
169.                     'If Team A has more mvp points then joins Team B
170.                     If TeamA.GetMVPPoints > TeamB.GetMVPPoints Then
171.                         TeamB.AddPlayerToSub(MALPlayerList(Subs(Count)), (MALPlayerList(Subs(Count)).GetNumberOfMVPPoints / 5))
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                            'If Team B has more mvp points then joins Team A
                        Else
                            TeamA.AddPlayerToSub(MALPlayerList(Subs(Count)), (MALPlayerList(Subs(Count)).GetNumberOfMVPPoints / 5))
                        End If
                    End If
                Next
            End Sub
            Sub FindRandomPlayer()
                Dim NumberOfPlayers As String
                Dim indexofprefferedposition As Integer
                'Finds player to play each position that doesn't have the correct number of player
                'Loops around each position
                For count = 0 To 6
                    'Number of players for the position
                    NumberOfPlayers = CInt(NumberOfPlayersPerPosition(count, 1)) + CInt(NumberOfPlayersPerPosition(count, 2)) + CInt(N
    umberOfPlayersPerPosition(count, 3)) + CInt(NumberOfPlayersPerPosition(count, 4)) + CInt(NumberOfPlayersPerPosition(count, 5)) + CInt
    (NumberOfPlayersPerPosition(count, 6))
                    'The number of player is less than the number needed
                    If NumberOfPlayers < NumberOfPlayersNeededPerPosition(count) Then
                        'Loops around each player
                        For playerindex = 0 To MALPlayerList.Count - 1
                            'Checks they haven't been asigned a position
                            If MALPlayerList(playerindex).GetPosition = "" And MALPlayerList(playerindex).GetFavouritePosition <> Posi
    tions(count) Then
                                'Finds the index of the players preferred position
                                indexofprefferedposition = GetIndexOfPosition(MALPlayerList(playerindex).GetFavouritePosition)
                                'Checks that the players preferred position has more than enough players
                                If NumberOfPlayersPerPosition(indexofprefferedposition, 1) > NumberOfPlayersNeededPerPosition(indexofp
    refferedposition) Then

                                    'The position only needs 1 player
                                    'Checks which player has played the least in their preferred position and records them
                                    If NumberOfPlayersNeededPerPosition(count) - NumberOfPlayers = 1 Then
                                        If NumberOfPlayersPerPosition(count, 7) = 0 Then
                                            NumberOfPlayersPerPosition(count, 7) += 1
                                            NumberOfPlayersPerPosition(indexofprefferedposition, 1) -= 1
                                            ReplacementPlayerTwoAtTheMoment(count, 1) = playerindex
                                        Else
                                            If MALPlayerList(count).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(Replacemen
    tPlayerTwoAtTheMoment(count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                                                NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerTwoAtTheM
    oment(count, 1)).GetFavouritePosition), 1) += 1
                                                NumberOfPlayersPerPosition(indexofprefferedposition, 1) -= 1
                                                ReplacementPlayerTwoAtTheMoment(count, 1) = playerindex
                                            End If

                                        End If
                                        'They need two players for position
                                        'Checks which player has played the least in their preferred position
                                    ElseIf NumberOfPlayersNeededPerPosition(count) - NumberOfPlayers = 2 Then
                                        If NumberOfPlayersPerPosition(count, 7) = 0 Then
                                            NumberOfPlayersPerPosition(count, 7) += 1
                                            NumberOfPlayersPerPosition(indexofprefferedposition, 1) -= 1
                                            ReplacementPlayerTwoAtTheMoment(count, 1) = playerindex
                                        ElseIf NumberOfPlayersPerPosition(count, 7) = 1 Then
                                            If MALPlayerList(count).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(Replacemen
    tPlayerTwoAtTheMoment(count, 1)).GetNumberOfTimesPlayedPreferredPosition Then

                                                NumberOfPlayersPerPosition(indexofprefferedposition, 1) -= 1
                                                ReplacementPlayerTwoAtTheMoment(count, 2) = ReplacementPlayerTwoAtTheMoment(count, 1)

                                                ReplacementPlayerTwoAtTheMoment(count, 1) = playerindex
                                            End If
                                        Else
                                            If MALPlayerList(count).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(Replacemen
    tPlayerTwoAtTheMoment(count, 2)).GetNumberOfTimesPlayedPreferredPosition Then
                                                If MALPlayerList(count).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(Replac
    ementPlayerTwoAtTheMoment(count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
                                                    NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerTwoAt
    TheMoment(count, 2)).GetFavouritePosition), 1) += 1
                                                    NumberOfPlayersPerPosition(indexofprefferedposition, 1) -= 1
                                                    ReplacementPlayerTwoAtTheMoment(count, 2) = ReplacementPlayerTwoAtTheMoment(count,
     1)
                                                    ReplacementPlayerTwoAtTheMoment(count, 1) = playerindex
                                                Else
                                                    NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerTwoAt
    TheMoment(count, 2)).GetFavouritePosition), 1) += 1
                                                    NumberOfPlayersPerPosition(indexofprefferedposition, 1) -= 1
                                                    ReplacementPlayerTwoAtTheMoment(count, 2) = playerindex
                                                End If
                                            End If
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                                    End If
                                End If
                            End If
                        End If
                    Next
                End If
                'Sets the players position to their playing position
                If NumberOfPlayersPerPosition(count, 7) = 1 Then
                    MALPlayerList(ReplacementPlayerTwoAtTheMoment(count, 1)).SetPosition(Positions(count))
                ElseIf NumberOfPlayersPerPosition(count, 7) = 2 Then
                    MALPlayerList(ReplacementPlayerTwoAtTheMoment(count, 1)).SetPosition(Positions(count))
                    MALPlayerList(ReplacementPlayerTwoAtTheMoment(count, 2)).SetPosition(Positions(count))
                End If
            Next
        End Sub
        Sub AssignPositions()
            'Loops through each position
            For Count = 0 To 6
                'If the position has less than 2 player
                If NumberOfPlayersPerPosition(Count, 1) < 3 Then
                    'Loops through each player
                    For playerindex = 0 To MALPlayerList.Count - 1
                        'If they haven't been assigned position
                        If MALPlayerList(playerindex).GetPosition = "" Then
                            'If their favourite position is the same as position
                            If MALPlayerList(playerindex).GetFavouritePosition = Positions(Count) Then
                                'Sets players position to their favourite position
                                MALPlayerList(playerindex).SetPosition(Positions(Count))
                            End If
                        End If
                    Next
                End If
            Next
            'Loops through each position
            For Count = 0 To 6
                'If position has more then 2 players then decides who is sub
                If NumberOfPlayersPerPosition(Count, 1) > 2 Then
                    DecideSubs(Count)
                End If
            Next

```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
281.            End Sub
282.            Function FitsSquad()
283.                'Checks whether the correct number of players for each position
284.                Dim Validplayers As Boolean
285.                Validplayers = True
286.                'Loops through each position
287.                For Count = 0 To 6
288.                    'If position has less players than needed then returns false
289.                    If (CInt(NumberOfPlayersPerPosition(Count, 1)) + CInt(NumberOfPlayersPerPosition(Count, 2)) + CInt(NumberOfPlayers
    PerPosition(Count, 3))) + (CInt(NumberOfPlayersPerPosition(Count, 4))) + (CInt(NumberOfPlayersPerPosition(Count, 5))) + (CInt(NumberO
    fPlayersPerPosition(Count, 6))) + (CInt(NumberOfPlayersPerPosition(Count, 7))) < NumberOfPlayersNeededPerPosition(Count) Then
290.                        Return False
291.                    End If
292.                Next
293.                Return True
294.            End Function
295.            Sub FindReplacementPlayersThirdPosition()
296.                Dim PrimaryPositionindex, secondarypositionindex As Integer
297.                Dim NumberOfPlayersCurrently As Integer
298.                'Loops through each postion
299.                For Count = 0 To 6
300.                    NumberOfPlayersPerPosition(Count, 3) = 0
301.                    NumberOfPlayersCurrently = NumberOfPlayersPerPosition(Count, 1) + NumberOfPlayersPerPosition(Count, 2)
302.                    'If position doesn't has enough players
303.                    If NumberOfPlayersCurrently < NumberOfPlayersNeededPerPosition(Count) Then
304.                        'Loops through each player
305.                        For playerindex = 0 To MALPlayerList.Count - 1
306.                            If MALPlayerList(playerindex).GetThirdFavouritePosition = Positions(Count) Then
307.                                'Gets index of player favourite and second favourite position
308.                                PrimaryPositionindex = GetIndexOfPosition(MALPlayerList(playerindex).GetFavouritePosition)
309.                                secondarypositionindex = GetIndexOfPosition(MALPlayerList(playerindex).GetSecondFavouritePosition)
310.                                If NumberOfPlayersPerPosition(PrimaryPositionindex, 1) + NumberOfPlayersPerPosition(PrimaryPositionind
    ex, 2) > NumberOfPlayersNeededPerPosition(PrimaryPositionindex) Then
311.                                    'If player hasn't been asigned position
312.                                    If MALPlayerList(playerindex).GetPosition = "" Then
313.                                        'Position only needs one player
314.                                        If NumberOfPlayersNeededPerPosition(Count) - NumberOfPlayersCurrently = 1 Then
315.                                            'Finds which players has played the least number of times in preferred position
316.                                            If NumberOfPlayersPerPosition(Count, 3) = 0 Then
317.                                                NumberOfPlayersPerPosition(Count, 3) += 1
318.                                                ReplacementPlayerTwoAtTheMoment(Count, 1) = playerindex
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                                                NumberOfPlayersPerPosition(PrimaryPositionindex, 1) -= 1
                                            Else
                                                If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(
    ReplacementPlayerTwoAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
                                                    NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerTwoAt
    TheMoment(Count, 1)).GetFavouritePosition), 1) += 1
                                                    ReplacementPlayerTwoAtTheMoment(Count, 1) = playerindex
                                                    NumberOfPlayersPerPosition(PrimaryPositionindex, 1) -= 1
                                                End If
                                            End If
                                            'The position needs two players
                                        ElseIf NumberOfPlayersNeededPerPosition(Count) - NumberOfPlayersCurrently = 2 Then
                                            If NumberOfPlayersPerPosition(Count, 3) = 0 Then
                                                NumberOfPlayersPerPosition(Count, 3) += 1
                                                ReplacementPlayerTwoAtTheMoment(Count, 1) = playerindex
                                                NumberOfPlayersPerPosition(PrimaryPositionindex, 1) -= 1
                                            ElseIf NumberOfPlayersPerPosition(Count, 3) = 1 Then
                                                NumberOfPlayersPerPosition(Count, 3) += 1
                                                If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(
    ReplacementPlayerTwoAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
                                                    ReplacementPlayerTwoAtTheMoment(Count, 2) = ReplacementPlayerTwoAtTheMoment(Count,
     1)
                                                    ReplacementPlayerTwoAtTheMoment(Count, 1) = playerindex
                                                    NumberOfPlayersPerPosition(PrimaryPositionindex, 1) -= 1
                                                Else
                                                    ReplacementPlayerTwoAtTheMoment(Count, 2) = playerindex
                                                    NumberOfPlayersPerPosition(PrimaryPositionindex, 1) -= 1
                                                End If
                                            ElseIf MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(
    ReplacementPlayerTwoAtTheMoment(Count, 2)).GetNumberOfTimesPlayedPreferredPosition Then
                                                If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(
    ReplacementPlayerTwoAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
                                                    NumberOfPlayersPerPosition(PrimaryPositionindex, 1) -= 1
                                                    NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerTwoAt
    TheMoment(Count, 2)).GetFavouritePosition), 1) += 1
                                                    ReplacementPlayerTwoAtTheMoment(Count, 2) = ReplacementPlayerTwoAtTheMoment(Count,
     1)
                                                    ReplacementPlayerTwoAtTheMoment(Count, 1) = playerindex
                                                Else
                                                    NumberOfPlayersPerPosition(PrimaryPositionindex, 1) -= 1
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                                                    NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerTwoAt
    TheMoment(Count, 2)).GetFavouritePosition), 1) += 1
                                                    ReplacementPlayerTwoAtTheMoment(Count, 2) = playerindex
                                                End If
                                            End If
                                        End If
                                    End If
                                End If
                            End If
                        Next
                    End If
                    'Assigns the players selected their position
                    If NumberOfPlayersPerPosition(Count, 3) = 1 Then
                        MALPlayerList(ReplacementPlayerTwoAtTheMoment(Count, 1)).SetPosition(Positions(Count))
                    ElseIf NumberOfPlayersPerPosition(Count, 3) = 2 Then
                        MALPlayerList(ReplacementPlayerTwoAtTheMoment(Count, 1)).SetPosition(Positions(Count))
                        MALPlayerList(ReplacementPlayerTwoAtTheMoment(Count, 2)).SetPosition(Positions(Count))

                    End If
                Next
            End Sub
            Sub FindReplacementPlayersSecondrayPosition()
                Dim PlayersIndexForPosition(2) As String

                Dim BetterPlayer, BetterPlayer2 As Boolean
                Dim indexofpreferredposition As Integer

                'Second FavouritePosition
                'For each player
                For Positionindex = 0 To 6
                    'If the position doesn't have enough player
                    If NumberOfPlayersPerPosition(Positionindex, 1) < NumberOfPlayersNeededPerPosition(Positionindex) Then

                        'Loops through each player
                        For playerindex = 0 To MALPlayerList.Count - 1
                            'Gets the index of the players preferred position
                            indexofpreferredposition = GetIndexOfPosition(MALPlayerList(playerindex).GetFavouritePosition)
                            'If players second favourite position is the current position
                            If MALPlayerList(playerindex).GetSecondFavouritePosition = Positions(Positionindex) Then
                                'If the players favourite position has too many players
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
390.                                 If NumberOfPlayersPerPosition(indexofpreferredposition, 1) > NumberOfPlayersNeededPerPosition(indexofp
    referredposition) Then
391.                                     'If the player needs one position
392.                                     If NumberOfPlayersNeededPerPosition(Positionindex) - NumberOfPlayersPerPosition(Positionindex, 1)
    = 1 Then
393.                                         If NumberOfPlayersPerPosition(Positionindex, 2) = 0 Then
394.                                             NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
395.                                             ReplacementPlayerAtTheMoment(Positionindex, 1) = playerindex
396.                                             NumberOfPlayersPerPosition(Positionindex, 2) += 1
397.                                         Else
398.                                             'Finds if the player should be the replacement player over the other player
399.                                             BetterPlayer = BetterReplacementplayer(playerindex, ReplacementPlayerAtTheMoment(Positioni
    ndex, 1), indexofpreferredposition)
400.                                             If BetterPlayer Then
401.                                                 'Records players details
402.                                                 NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerAtTheMome
    nt(Positionindex, 1)).GetFavouritePosition), 1) -= 1
403.                                                 NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
404.                                                 ReplacementPlayerAtTheMoment(Positionindex, 1) = playerindex
405.                                             End If
406.                                         End If
407.                                         'Two players are needed
408.                                     ElseIf NumberOfPlayersNeededPerPosition(Positionindex) - NumberOfPlayersPerPosition(Positionindex,
     1) = 2 Then
409.                                         If NumberOfPlayersPerPosition(Positionindex, 2) = 0 Then
410.                                             NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
411.                                             ReplacementPlayerAtTheMoment(Positionindex, 1) = playerindex
412.                                             NumberOfPlayersPerPosition(Positionindex, 2) += 1
413.                                         ElseIf NumberOfPlayersPerPosition(Positionindex, 2) = 1 Then
414.                                             'Find which player should play their second favourite position
415.                                             NumberOfPlayersPerPosition(Positionindex, 2) += 1
416.                                             BetterPlayer = BetterReplacementplayer(playerindex, ReplacementPlayerAtTheMoment(Positioni
    ndex, 1), indexofpreferredposition)
417.                                             NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
418.                                             If BetterPlayer Then
419.
420.                                                 ReplacementPlayerAtTheMoment(Positionindex, 2) = ReplacementPlayerAtTheMoment(Position
    index, 1)
421.                                                 ReplacementPlayerAtTheMoment(Positionindex, 1) = playerindex
422.                                             Else
423.                                                 ReplacementPlayerAtTheMoment(Positionindex, 2) = playerindex
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                                            End If
                                        Else
                                            BetterPlayer = BetterReplacementplayer(playerindex, ReplacementPlayerAtTheMoment(Positioni
    ndex, 2), indexofpreferredposition)
                                            If BetterPlayer Then
                                                BetterPlayer2 = BetterReplacementplayer(playerindex, ReplacementPlayerAtTheMoment(Posi
    tionindex, 1), indexofpreferredposition)
                                                If BetterPlayer2 Then
                                                    ReplacementPlayerAtTheMoment(Positionindex, 2) = ReplacementPlayerAtTheMoment(Posi
    tionindex, 1)
                                                    ReplacementPlayerAtTheMoment(Positionindex, 1) = playerindex
                                                    NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerAtThe
    Moment(Positionindex, 2)).GetFavouritePosition), 1) = +1
                                                    NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                Else
                                                    NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerAtThe
    Moment(Positionindex, 2)).GetFavouritePosition), 1) = +1
                                                    ReplacementPlayerAtTheMoment(Positionindex, 2) = Positionindex
                                                    NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                End If

                                            End If
                                        End If
                                    End If
                                End If
                            End If
                        Next
                    End If
                    'Sets the players position to their new position
                    If NumberOfPlayersPerPosition(Positionindex, 2) = 1 Then
                        MALPlayerList(ReplacementPlayerAtTheMoment(Positionindex, 1)).SetPosition(Positions(Positionindex))
                    ElseIf NumberOfPlayersPerPosition(Positionindex, 2) = 2 Then
                        MALPlayerList(ReplacementPlayerAtTheMoment(Positionindex, 1)).SetPosition(Positions(Positionindex))
                        MALPlayerList(ReplacementPlayerAtTheMoment(Positionindex, 2)).SetPosition(Positions(Positionindex))
                    End If
                Next
            End Sub
            Function GetIndexOfPosition(ByVal CompareValue) As String
                'Gets index of position
                For Count = 0 To 6
                    If Positions(Count) = CompareValue Then
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
460.                     Return Count
461.                 End If
462.             Next
463.             Return -1
464.         End Function
465.         Function BetterReplacementplayer(ByVal newplayerindex As String, ByVal orginalplayerindex As String, ByVal indexofpreferre
    dposition As Integer)
466.             Dim indexofthirdfavouriteposition As Integer
467.             indexofthirdfavouriteposition = GetIndexOfPosition(MALPlayerList(newplayerindex).GetThirdFavouritePosition)
468.             'Checks number of players for players third favourite position
469.             If NumberOfPlayersPerPosition(indexofthirdfavouriteposition, 1) > NumberOfPlayersPerPosition(GetIndexOfPosition(MALPla
    yerList(orginalplayerindex).GetThirdFavouritePosition), 1) Then
470.                 Return True
471.             ElseIf NumberOfPlayersPerPosition(indexofthirdfavouriteposition, 1) = NumberOfPlayersPerPosition(GetIndexOfPosition(MA
    LPlayerList(orginalplayerindex).GetFavouritePosition), 1) Then
472.                 'Checks number of players for players first favourite position
473.                 If NumberOfPlayersPerPosition(indexofpreferredposition, 1) > NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlay
    erList(orginalplayerindex).GetFavouritePosition), 1) Then
474.                     Return True
475.                 ElseIf NumberOfPlayersPerPosition(indexofpreferredposition, 1) = NumberOfPlayersPerPosition(GetIndexOfPosition(MAL
    PlayerList(orginalplayerindex).GetFavouritePosition), 1) Then
476.                     Return False
477.                 Else
478.                     'Checks the number of times the player has player their preferred position
479.                     If MALPlayerList(newplayerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(orginalplayerindex).G
    etNumberOfTimesPlayedPreferredPosition Then
480.                         Return True
481.                     Else
482.                         Return False
483.                     End If
484.                 End If
485.             Else
486.                 Return False
487.             End If
488.         End Function
489.         Sub FindArea3Replacement()
490.             Dim NumberOfPlayersForPosition As Integer
491.             Dim PositionArea, playerspositionarea As String
492.             Dim indexofpreferredposition, indexofthirdposition As Integer
493.             'Loops through each position
494.             For Count = 0 To 6
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                    NumberOfPlayersForPosition = CInt(NumberOfPlayersPerPosition(Count, 1)) + CInt(NumberOfPlayersPerPosition(Count, 2
    )) + CInt(NumberOfPlayersPerPosition(Count, 3)) + CInt(NumberOfPlayersPerPosition(Count, 4))
                    'Checks the number of players for that position if its less than the required number
                    If NumberOfPlayersForPosition < NumberOfPlayersNeededPerPosition(Count) Then
                        'Works out area of position
                        If Count = 0 Or Count = 1 Then
                            PositionArea = "DEF"
                        ElseIf Count = 2 Or Count = 3 Or Count = 4 Then
                            PositionArea = "MID"
                        Else
                            PositionArea = "ATT"
                        End If
                        'Loops through each player
                        For playerindex = 0 To MALPlayerList.Count - 1
                            'If they haven't been asigned a position
                            If MALPlayerList(playerindex).GetPosition = "" Then
                                'Gets index of players preferred position
                                indexofthirdposition = GetIndexOfPosition(MALPlayerList(playerindex).GetThirdFavouritePosition)
                                indexofpreferredposition = GetIndexOfPosition(MALPlayerList(playerindex).GetFavouritePosition)
                                'Works out area of players 3rd favourite position
                                If indexofthirdposition = 0 Or indexofthirdposition = 1 Then
                                    playerspositionarea = "DEF"
                                ElseIf indexofpreferredposition = 2 Or indexofthirdposition = 3 Or indexofthirdposition = 4 Then
                                    playerspositionarea = "MID"
                                Else
                                    playerspositionarea = "ATT"
                                End If
                                'If the positions area and the players third favourite position area match
                                If PositionArea = playerspositionarea Then
                                    'If the players preferred position has enough
                                    If CInt(NumberOfPlayersPerPosition(indexofpreferredposition, 1)) + CInt(NumberOfPlayersPerPosition
    (indexofpreferredposition, 2)) + CInt(NumberOfPlayersPerPosition(indexofpreferredposition, 3)) > NumberOfPlayersNeededPerPosition(ind
    exofpreferredposition) Then
                                        'Checks they haven't been asigned a position
                                        If MALPlayerList(playerindex).GetPosition = "" Then
                                            If NumberOfPlayersNeededPerPosition(Count) - NumberOfPlayersForPosition = 1 Then
                                                If NumberOfPlayersPerPosition(Count, 6) = 0 Then
                                                    NumberOfPlayersPerPosition(Count, 6) += 1
                                                    NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                    ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
                                                Else
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
533.                                                 'Checks to see who has played more in their favourite position
534.                                                 If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerL
    ist(ReplacementPlayerThreeAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
535.                                                     NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
536.                                                     NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerT
    hreeAtTheMoment(Count, 1)).GetFavouritePosition), 1) += 1
537.                                                     ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
538.
539.                                                 End If
540.                                             End If
541.                                             'Needs two players
542.                                             'Checks to see who has played their preferred positon more
543.                                         ElseIf NumberOfPlayersNeededPerPosition(Count) - NumberOfPlayersForPosition = 2 Then
544.                                             If NumberOfPlayersPerPosition(Count, 6) = 0 Then
545.                                                 NumberOfPlayersPerPosition(Count, 6) += 1
546.                                                 NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
547.                                                 ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
548.                                             ElseIf NumberOfPlayersPerPosition(Count, 6) = 1 Then
549.                                                 NumberOfPlayersPerPosition(Count, 6) += 1
550.                                                 If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerL
    ist(ReplacementPlayerThreeAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
551.                                                     NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
552.                                                     ReplacementPlayerThreeAtTheMoment(Count, 2) = ReplacementPlayerThreeAtTheMomen
    t(Count, 1)
553.                                                     ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
554.                                                 Else
555.                                                     NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
556.                                                     ReplacementPlayerThreeAtTheMoment(Count, 2) = playerindex
557.                                                 End If
558.                                             Else
559.                                                 If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerL
    ist(ReplacementPlayerThreeAtTheMoment(Count, 2)).GetNumberOfTimesPlayedPreferredPosition Then
560.                                                     If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPla
    yerList(ReplacementPlayerThreeAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
561.                                                         NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPla
    yerThreeAtTheMoment(Count, 2)).GetFavouritePosition), 1) += 2
562.                                                         NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
563.                                                         ReplacementPlayerThreeAtTheMoment(Count, 2) = ReplacementPlayerThreeAtTheM
    oment(Count, 1)
564.                                                         ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
565.                                                     Else
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                                                                NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPla
    yerThreeAtTheMoment(Count, 2)).GetFavouritePosition), 1) += 2
                                                                NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                                ReplacementPlayerThreeAtTheMoment(Count, 2) = playerindex
                                                            End If
                                                        End If
                                                    End If
                                                End If
                                            End If
                                        End If
                                    End If
                                Next

                            End If
                            'Assigns the players selected to the position
                            If NumberOfPlayersPerPosition(Count, 6) = 1 Then
                                MALPlayerList(ReplacementPlayerThreeAtTheMoment(Count, 1)).SetPosition(Positions(Count))
                            ElseIf NumberOfPlayersPerPosition(Count, 6) = 2 Then
                                MALPlayerList(ReplacementPlayerThreeAtTheMoment(Count, 1)).SetPosition(Positions(Count))
                                MALPlayerList(ReplacementPlayerThreeAtTheMoment(Count, 2)).SetPosition(Positions(Count))
                            End If
                        Next
                    End Sub
                    Sub FindArea2Replacement()
                        Dim NumberOfPlayersForPosition As Integer
                        Dim PositionArea, playerspositionarea As String
                        Dim indexofpreferredposition, indexofsecondposition As Integer
                        'Loops through each position
                        For Count = 0 To 6
                            NumberOfPlayersForPosition = CInt(NumberOfPlayersPerPosition(Count, 1)) + CInt(NumberOfPlayersPerPosition(Count, 2
    )) + CInt(NumberOfPlayersPerPosition(Count, 3)) + CInt(NumberOfPlayersPerPosition(Count, 4))
                            'Checks if the position has enough players
                            If NumberOfPlayersForPosition < NumberOfPlayersNeededPerPosition(Count) Then
                                'Calculates the area of the pitch for the position that they are searching on
                                If Count = 0 Or Count = 1 Then
                                    PositionArea = "DEF"
                                ElseIf Count = 2 Or Count = 3 Or Count = 4 Then
                                    PositionArea = "MID"
                                Else
                                    PositionArea = "ATT"
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
605.                         End If
606.                         'Loops through each player
607.                         For playerindex = 0 To MALPlayerList.Count - 1
608.                             'Checks if player has been asigned position
609.                             If MALPlayerList(playerindex).GetPosition = "" Then
610.                                 'Get index of first and second position
611.                                 indexofsecondposition = GetIndexOfPosition(MALPlayerList(playerindex).GetSecondFavouritePosition)
612.                                 indexofpreferredposition = GetIndexOfPosition(MALPlayerList(playerindex).GetFavouritePosition)
613.                                 'Get area of players second favourite position
614.                                 If indexofsecondposition = 0 Or indexofsecondposition = 1 Then
615.                                     playerspositionarea = "DEF"
616.                                 ElseIf indexofpreferredposition = 2 Or indexofsecondposition = 3 Or indexofsecondposition = 4 Then
617.                                     playerspositionarea = "MID"
618.                                 Else
619.                                     playerspositionarea = "ATT"
620.                                 End If
621.                                 'If positions area equals the players second favourite area
622.                                 If PositionArea = playerspositionarea Then
623.                                     'If player favourite position has too many players
624.                                     If CInt(NumberOfPlayersPerPosition(indexofpreferredposition, 1)) + CInt(NumberOfPlayersPerPosition
    (indexofpreferredposition, 2)) + CInt(NumberOfPlayersPerPosition(indexofpreferredposition, 3)) > NumberOfPlayersNeededPerPosition(ind
    exofpreferredposition) Then
625.                                         'If position needs one player
626.                                         If NumberOfPlayersNeededPerPosition(Count) - NumberOfPlayersForPosition = 1 Then
627.                                             If NumberOfPlayersPerPosition(Count, 5) = 0 Then
628.                                                 NumberOfPlayersPerPosition(Count, 5) += 1
629.                                                 NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
630.                                                 ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
631.                                             Else
632.                                                 'Finds which player has played preferred position more times
633.                                                 If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(
    ReplacementPlayerThreeAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
634.                                                     NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
635.                                                     NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerThree
    AtTheMoment(Count, 1)).GetFavouritePosition), 1) += 1
636.                                                     ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
637.
638.                                                 End If
639.                                             End If
640.                                             'They need two players
641.                                         ElseIf NumberOfPlayersNeededPerPosition(Count) - NumberOfPlayersForPosition = 2 Then
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                                            If NumberOfPlayersPerPosition(Count, 5) = 0 Then
                                                NumberOfPlayersPerPosition(Count, 5) += 1
                                                NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
                                                'Finds out which player has played preferred position more times
                                            ElseIf NumberOfPlayersPerPosition(Count, 5) = 1 Then
                                                NumberOfPlayersPerPosition(Count, 5) += 1
                                                If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(
    ReplacementPlayerThreeAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
                                                    NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                    ReplacementPlayerThreeAtTheMoment(Count, 2) = ReplacementPlayerThreeAtTheMoment(Co
    unt, 1)
                                                    ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
                                                Else
                                                    NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                    ReplacementPlayerThreeAtTheMoment(Count, 2) = playerindex
                                                End If
                                            Else
                                                If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(
    ReplacementPlayerThreeAtTheMoment(Count, 2)).GetNumberOfTimesPlayedPreferredPosition Then
                                                    If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerL
    ist(ReplacementPlayerThreeAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
                                                        NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerT
    hreeAtTheMoment(Count, 2)).GetFavouritePosition), 1) += 2
                                                        NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                        ReplacementPlayerThreeAtTheMoment(Count, 2) = ReplacementPlayerThreeAtTheMomen
    t(Count, 1)
                                                        ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
                                                    Else
                                                        NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerT
    hreeAtTheMoment(Count, 2)).GetFavouritePosition), 1) += 2
                                                        NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                        ReplacementPlayerThreeAtTheMoment(Count, 2) = playerindex
                                                    End If
                                                End If
                                            End If
                                        End If
                                    End If
                                End If
                            End If
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                        Next

                    End If
                    'Assigns players the new position
                    If NumberOfPlayersPerPosition(Count, 5) = 1 Then
                        MALPlayerList(ReplacementPlayerThreeAtTheMoment(Count, 1)).SetPosition(Positions(Count))
                    ElseIf NumberOfPlayersPerPosition(Count, 5) = 2 Then
                        MALPlayerList(ReplacementPlayerThreeAtTheMoment(Count, 1)).SetPosition(Positions(Count))
                        MALPlayerList(ReplacementPlayerThreeAtTheMoment(Count, 2)).SetPosition(Positions(Count))
                    End If

                Next
            End Sub
            Sub FindAreaReplacement()
                Dim NumberOfPlayersForPosition As Integer
                Dim PositionArea, playerspositionarea As String
                Dim indexofpreferredposition As Integer
                'Loops through each position
                For Count = 0 To 6
                    NumberOfPlayersForPosition = CInt(NumberOfPlayersPerPosition(Count, 1)) + CInt(NumberOfPlayersPerPosition(Count, 2
    )) + CInt(NumberOfPlayersPerPosition(Count, 3))
                    'Checks the that the positions has too little players
                    If NumberOfPlayersForPosition < NumberOfPlayersNeededPerPosition(Count) Then
                        'Finds area of position
                        If Count = 0 Or Count = 1 Then
                            PositionArea = "DEF"
                        ElseIf Count = 2 Or Count = 3 Or Count = 4 Then
                            PositionArea = "MID"
                        Else
                            PositionArea = "ATT"
                        End If
                        'Loops through each player
                        For playerindex = 0 To MALPlayerList.Count - 1
                            'Checks that they haven't been assigned a position
                            If MALPlayerList(playerindex).GetPosition = "" Then
                                indexofpreferredposition = GetIndexOfPosition(MALPlayerList(playerindex).GetFavouritePosition)
                                'Finds area of preferred position
                                If indexofpreferredposition = 0 Or indexofpreferredposition = 1 Then
                                    playerspositionarea = "DEF"
                                ElseIf indexofpreferredposition = 2 Or indexofpreferredposition = 3 Or indexofpreferredposition = 4 Th
    en
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                                    playerspositionarea = "MID"
                                Else
                                    playerspositionarea = "ATT"
                                End If
                                'If positions area equals the area of players preferred position
                                If PositionArea = playerspositionarea Then

                                    'If players favourite position has too many players
                                    If CInt(NumberOfPlayersPerPosition(indexofpreferredposition, 1)) + CInt(NumberOfPlayersPerPosition
(indexofpreferredposition, 2)) + CInt(NumberOfPlayersPerPosition(indexofpreferredposition, 3)) > NumberOfPlayersNeededPerPosition(ind
exofpreferredposition) Then
                                        'Position only needs one player
                                        If NumberOfPlayersNeededPerPosition(Count) - NumberOfPlayersForPosition = 1 Then
                                            If NumberOfPlayersPerPosition(Count, 4) = 0 Then
                                                NumberOfPlayersPerPosition(Count, 4) += 1
                                                NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
                                            Else
                                                'Finds which player has played preferred position the most
                                                If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(
ReplacementPlayerThreeAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
                                                    NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                    NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerThree
AtTheMoment(Count, 1)).GetFavouritePosition), 1) += 1
                                                    ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex

                                                End If
                                            End If
                                            'Position needs two players
                                        ElseIf NumberOfPlayersNeededPerPosition(Count) - NumberOfPlayersForPosition = 2 Then
                                            'Finds which player has played preferred position the most
                                            If NumberOfPlayersPerPosition(Count, 4) = 0 Then
                                                NumberOfPlayersPerPosition(Count, 4) += 1
                                                NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
                                            ElseIf NumberOfPlayersPerPosition(Count, 4) = 1 Then
                                                NumberOfPlayersPerPosition(Count, 4) += 1
                                                If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(
ReplacementPlayerThreeAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
                                                    NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                                                ReplacementPlayerThreeAtTheMoment(Count, 2) = ReplacementPlayerThreeAtTheMoment(Co
    unt, 1)
                                                ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
                                            Else
                                                NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                ReplacementPlayerThreeAtTheMoment(Count, 2) = playerindex
                                            End If
                                        Else
                                            If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerList(
    ReplacementPlayerThreeAtTheMoment(Count, 2)).GetNumberOfTimesPlayedPreferredPosition Then
                                                If MALPlayerList(playerindex).GetNumberOfTimesPlayedPreferredPosition > MALPlayerL
    ist(ReplacementPlayerThreeAtTheMoment(Count, 1)).GetNumberOfTimesPlayedPreferredPosition Then
                                                    NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerT
    hreeAtTheMoment(Count, 2)).GetFavouritePosition), 1) += 2
                                                    NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                    ReplacementPlayerThreeAtTheMoment(Count, 2) = ReplacementPlayerThreeAtTheMomen
    t(Count, 1)
                                                    ReplacementPlayerThreeAtTheMoment(Count, 1) = playerindex
                                                Else
                                                    NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(ReplacementPlayerT
    hreeAtTheMoment(Count, 2)).GetFavouritePosition), 1) += 2
                                                    NumberOfPlayersPerPosition(indexofpreferredposition, 1) -= 1
                                                    ReplacementPlayerThreeAtTheMoment(Count, 2) = playerindex
                                                End If
                                            End If
                                        End If
                                    End If
                                End If
                            End If
                        Next
                    End If
                    'Assingns the position to the selected player
                    If NumberOfPlayersPerPosition(Count, 4) = 1 Then
                        MALPlayerList(ReplacementPlayerThreeAtTheMoment(Count, 1)).SetPosition(Positions(Count))
                    ElseIf NumberOfPlayersPerPosition(Count, 4) = 2 Then
                        MALPlayerList(ReplacementPlayerThreeAtTheMoment(Count, 1)).SetPosition(Positions(Count))
                        MALPlayerList(ReplacementPlayerThreeAtTheMoment(Count, 2)).SetPosition(Positions(Count))
                    End If
                Next
            End Sub
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
785.          Sub AssignPlayersToATeam()
786.              Dim PlayersIndexForPosition(2) As String
787.              Dim Player1MVPPoints, Player2MVPPoints As Integer
788.              Dim Player1PreferredPosition, Player2PreferredPosition As Boolean
789.              Dim NumberOfPlayer As Integer
790.              For Count = 0 To 6
791.                  NumberOfPlayer = 0
792.                  'Finds player who are playing each position
793.                  'Loops through each player
794.                  For playerindex = 0 To MALPlayerList.Count - 1
795.                      'If their position matches the position
796.                      If MALPlayerList(playerindex).GetPosition = Positions(Count) Then
797.                          NumberOfPlayer += 1
798.                          'Stores index of players position
799.                          PlayersIndexForPosition(NumberOfPlayer) = playerindex
800.                          'Checks whether the player is playing their preferred position
801.                          If MALPlayerList(playerindex).GetPosition = MALPlayerList(playerindex).GetFavouritePosition Then
802.                              If NumberOfPlayer = 1 Then
803.                                  Player1PreferredPosition = True
804.                              ElseIf NumberOfPlayer = 2 Then
805.                                  Player2PreferredPosition = True
806.                              End If
807.                          End If
808.                      End If
809.                  Next
810.                  If NumberOfPlayer = 2 Then
811.                      'Calculates the number of mvp points for players depending on whether they are playing their preferred positio
    n
812.                      If MALPlayerList(PlayersIndexForPosition(1)).GetPosition = MALPlayerList(PlayersIndexForPosition(1)).GetSecond
    FavouritePosition Then
813.                          Player1MVPPoints = MALPlayerList(PlayersIndexForPosition(1)).GetNumberOfMVPPoints() / 2
814.                      ElseIf MALPlayerList(PlayersIndexForPosition(1)).GetPosition = MALPlayerList(PlayersIndexForPosition(1)).GetTh
    irdFavouritePosition Then
815.                          Player1MVPPoints = MALPlayerList(PlayersIndexForPosition(1)).GetNumberOfMVPPoints() / 3
816.                      ElseIf MALPlayerList(PlayersIndexForPosition(1)).GetPosition = MALPlayerList(PlayersIndexForPosition(1)).GetFa
    vouritePosition Then
817.                          Player1MVPPoints = MALPlayerList(PlayersIndexForPosition(1)).GetNumberOfMVPPoints()
818.                      Else
819.                          Player1MVPPoints = MALPlayerList(PlayersIndexForPosition(1)).GetNumberOfMVPPoints() / 4
820.                      End If
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                        'Calculates the number of mvp points for players depending on whether they are playing their preferred positio
    n
                        If MALPlayerList(PlayersIndexForPosition(2)).GetPosition = MALPlayerList(PlayersIndexForPosition(2)).GetSecond
    FavouritePosition Then
                            Player2MVPPoints = MALPlayerList(PlayersIndexForPosition(2)).GetNumberOfMVPPoints() / 2
                        ElseIf MALPlayerList(PlayersIndexForPosition(2)).GetPosition = MALPlayerList(PlayersIndexForPosition(2)).GetTh
    irdFavouritePosition Then
                            Player2MVPPoints = MALPlayerList(PlayersIndexForPosition(2)).GetNumberOfMVPPoints() / 3
                        ElseIf MALPlayerList(PlayersIndexForPosition(2)).GetPosition = MALPlayerList(PlayersIndexForPosition(2)).GetFa
    vouritePosition Then
                            Player2MVPPoints = MALPlayerList(PlayersIndexForPosition(2)).GetNumberOfMVPPoints()
                        Else
                            Player2MVPPoints = MALPlayerList(PlayersIndexForPosition(2)).GetNumberOfMVPPoints() / 4
                        End If
                        'Allocates players to team
                        'If team a has more players playing thier preferred position than team b
                        If TeamA.NumberPlayingPreferredPosition >= TeamB.NumberPlayingPreferredPosition Then
                            'If player 1 is playing preferred position and player 2 isn't playing preferred position
                            If Player1PreferredPosition = True And Player2PreferredPosition = False Then
                                'Player 2 joins team a and player 1 joins team b
                                TeamA.AddPlayerToTeam(MALPlayerList(PlayersIndexForPosition(2)), Player2MVPPoints, Player2PreferredPos
    ition)
                                TeamB.AddPlayerToTeam(MALPlayerList(PlayersIndexForPosition(1)), Player1MVPPoints, Player1PreferredPos
    ition)
                                'If player 2 is playing preferred position and player 1 isn't playing preferred position
                            ElseIf Player1PreferredPosition = False And Player2PreferredPosition = True Then
                                'Player 2 joins team B and player 1 joins team a
                                TeamB.AddPlayerToTeam(MALPlayerList(PlayersIndexForPosition(2)), Player2MVPPoints, Player2PreferredPos
    ition)
                                TeamA.AddPlayerToTeam(MALPlayerList(PlayersIndexForPosition(1)), Player1MVPPoints, Player1PreferredPos
    ition)
                            Else
                                'Decides which team the players join based upon mvp points
                                CalculateMVPPoints(Player1MVPPoints, Player2MVPPoints, Player1PreferredPosition, Player2PreferredPosit
    ion, PlayersIndexForPosition(1), PlayersIndexForPosition(2))
                            End If
                            'If team b has more players playing thier preferred position than team a
                        ElseIf TeamA.NumberPlayingPreferredPosition < TeamB.NumberPlayingPreferredPosition Then
                            'If player 1 is playing preferred position and player 2 isn't playing preferred position
                            If Player1PreferredPosition = True And Player2PreferredPosition = False Then
                                'Player 2 joins team B and player 1 joins team a
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                                TeamB.AddPlayerToTeam(MALPlayerList(PlayersIndexForPosition(2)), Player2MVPPoints, Player2PreferredPos
    ition)
                                TeamA.AddPlayerToTeam(MALPlayerList(PlayersIndexForPosition(1)), Player1MVPPoints, Player1PreferredPos
    ition)
                                'If player 2 is playing preferred position and player 1 isn't playing preferred position
                            ElseIf Player1PreferredPosition = False And Player2PreferredPosition = True Then
                                'Player 2 joins team a and player 1 joins team b
                                TeamA.AddPlayerToTeam(MALPlayerList(PlayersIndexForPosition(2)), Player2MVPPoints, Player2PreferredPos
    ition)
                                TeamB.AddPlayerToTeam(MALPlayerList(PlayersIndexForPosition(1)), Player1MVPPoints, Player1PreferredPos
    ition)
                            Else
                                'Decides which team the players join based upon mvp points
                                CalculateMVPPoints(Player1MVPPoints, Player2MVPPoints, Player1PreferredPosition, Player2PreferredPosit
    ion, PlayersIndexForPosition(1), PlayersIndexForPosition(2))
                            End If
                        Else
                            'Decides which team the players join based upon mvp points
                            CalculateMVPPoints(Player1MVPPoints, Player2MVPPoints, Player1PreferredPosition, Player2PreferredPosition,
     PlayersIndexForPosition(1), PlayersIndexForPosition(2))
                        End If
                        'If position only needs 1 player
                    ElseIf NumberOfPlayer = 1 Then
                        FloatingCentre = PlayersIndexForPosition(1)
                    End If
                Next
            End Sub
            Sub CalculateMVPPoints(ByVal Player1MVP As Integer, ByVal Player2MVP As Integer, ByVal Player1PreferredPos As Boolean, ByV
    al Player2PerferredPos As Boolean, ByVal Player1Index As String, ByVal Player2Index As String)
                'If team A has more mvp points than team b
                If TeamA.GetMVPPoints >= TeamB.GetMVPPoints Then
                    'If player 1 has more mvp points than player 2
                    If Player1MVP >= Player2MVP Then
                        'Adds player 2 to team a and player 1 to team b
                        TeamA.AddPlayerToTeam(MALPlayerList(Player2Index), Player2MVP, Player2PerferredPos)
                        TeamB.AddPlayerToTeam(MALPlayerList(Player1Index), Player1MVP, Player1PreferredPos)
                        'If player 2 has more mvp points than player 1
                    Else
                        'Adds player 1 to team a and player 2 to team b
                        TeamB.AddPlayerToTeam(MALPlayerList(Player2Index), Player2MVP, Player2PerferredPos)
                        TeamA.AddPlayerToTeam(MALPlayerList(Player1Index), Player1MVP, Player1PreferredPos)
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
887.                 End If
888.                 'If team b has more mvp points that team b
889.             Else
890.                 'If player 1 has more mvp points than player 2
891.                 If Player1MVP >= Player2MVP Then
892.                     'Player 2 joins team b and player 1 joins team a
893.                     TeamB.AddPlayerToTeam(MALPlayerList(Player2Index), Player2MVP, Player2PerferredPos)
894.                     TeamA.AddPlayerToTeam(MALPlayerList(Player1Index), Player1MVP, Player1PreferredPos)
895.                 Else
896.                     'Player 1 joins team b and player 2 joins team a
897.                     TeamA.AddPlayerToTeam(MALPlayerList(Player2Index), Player2MVP, Player2PerferredPos)
898.                     TeamB.AddPlayerToTeam(MALPlayerList(Player1Index), Player1MVP, Player1PreferredPos)
899.                 End If
900.             End If
901.         End Sub
902.
903.         Sub FindNumberOfPlayerForEachPosition()
904.             'Loops through each player and recond number of players for position
905.             For playerindex = 0 To MALPlayerList.Count - 1
906.                 NumberOfPlayersPerPosition(GetIndexOfPosition(MALPlayerList(playerindex).GetFavouritePosition), 1) += 1
907.             Next
908.             'For each position calculates whether right number of player
909.             For Count = 0 To 6
910.                 If NumberOfPlayersPerPosition(Count, 1) = NumberOfPlayersNeededPerPosition(Count) Then
911.                     NumberOfPositionsWithCorrectAmount = NumberOfPositionsWithCorrectAmount + 1
912.                 ElseIf NumberOfPlayersPerPosition(Count, 1) < NumberOfPlayersNeededPerPosition(Count) Then
913.                     NumberOfPositionsWithTooLittle = NumberOfPositionsWithTooLittle + 1
914.                 ElseIf NumberOfPlayersPerPosition(Count, 1) > NumberOfPlayersNeededPerPosition(Count) Then
915.                     NumberOfPositionsWithTooMany = NumberOfPositionsWithTooMany + 1
916.                 End If
917.             Next
918.         End Sub
919.         Sub DisplayTeams()
920.             Dim TeamAList As ArrayList
921.             Dim TeamBList As ArrayList
922.             Dim TeamASubs As ArrayList
923.             Dim TeamBSubs As ArrayList
924.             SessionSelection.Hide()
925.             SelectSessionLBL.Hide()
926.             'Shows labels which will show both teams
927.             LBL_TeamA.Show()
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
928.                LBL_TeamB.Show()
929.                LBLTeamAPlayers.Show()
930.                LBLTeamBPlayers.Show()
931.                'Gets the team and subs for each team
932.                TeamAList = TeamA.GetTeam
933.                TeamBList = TeamB.GetTeam
934.                TeamASubs = TeamA.GetSubs
935.                TeamBSubs = TeamB.GetSubs
936.                'Show the image of the netball court depending on number of players
937.                If MALPlayerList.Count >= 14 Then
938.                    P14.Show()
939.                ElseIf MALPlayerList.Count = 13 Then
940.                    P13.Show()
941.                ElseIf MALPlayerList.Count = 12 Then
942.                    P12.Show()
943.                ElseIf MALPlayerList.Count = 11 Then
944.                    P11.Show()
945.                ElseIf MALPlayerList.Count = 10 Then
946.                    P10.Show()
947.                ElseIf MALPlayerList.Count = 9 Then
948.                    P9.Show()
949.                End If
950.                'Adds Player and their position from Team A
951.                For Count = 0 To TeamAList.Count - 1
952.                    LBLTeamAPlayers.Text = LBLTeamAPlayers.Text + Environment.NewLine + TeamAList(Count).GetPosition + "  -  " & TeamA
    List(Count).GetFullName
953.                Next
954.                'Adds Player and their position from Team B
955.                For Count = 0 To TeamBList.Count - 1
956.                    LBLTeamBPlayers.Text = LBLTeamBPlayers.Text + Environment.NewLine + TeamBList(Count).GetPosition + "  -  " & TeamB
    List(Count).GetFullName
957.                Next
958.                'Adds Player for Team A Sub
959.                For Count = 0 To TeamA.NumberOfSums - 1
960.                    LBLTeamAPlayers.Text = LBLTeamAPlayers.Text + Environment.NewLine + "SUB" + "  -  " + TeamASubs(Count).GetFullName

961.                Next
962.                'Adds Player for Team B Sub
963.                For Count = 0 To TeamB.NumberOfSums - 1
964.                    LBLTeamBPlayers.Text = LBLTeamBPlayers.Text + Environment.NewLine + "SUB" + "  -  " & TeamBSubs(Count).GetFullName
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
965.                Next
966.                'If there is a floating centre then adds them to the label
967.                If FloatingCentre <> -1 Then
968.                    'Shows label
969.                    LBLFloatingCentre.Show()
970.                    LBLFloatingCentre.Text = "Floating Centre - " & MALPlayerList(FloatingCentre).GetFullName
971.                End If
972.                'Shows the save team button
973.                BTN_SaveTeam.Show()
974.            End Sub
975.            Sub DecideSubs(ByVal positionindex As String)
976.                Dim NumberOfOptions As Integer = 0
977.                Dim Options(NumberOfPlayersPerPosition(positionindex, 1)) As String
978.                Dim Holder(2, 2) As String
979.                Dim TempTimesHolder, TempIDHolder As Integer
980.                'Finds players who has that position has their favourite position
981.                For playerindex = 0 To MALPlayerList.Count - 1
982.                    If Positions(positionindex) = MALPlayerList(playerindex).GetFavouritePosition And MALPlayerList(playerindex).GetPo
    sition = "" Then
983.                        NumberOfOptions += 1
984.                        Options(NumberOfOptions) = playerindex
985.                    End If
986.                Next
987.                'Decides who should be sub by the number of time they have been sub
988.                'Loops through each option
989.                For count = 1 To NumberOfOptions
990.                    TempIDHolder = Options(count)
991.                    TempTimesHolder = MALPlayerList(TempIDHolder).GetNumberOfTimesSub
992.                    'If first player then stores the player details
993.                    If count = 1 Then
994.                        Holder(1, 1) = TempIDHolder
995.                        Holder(1, 2) = TempTimesHolder
996.                        'If second player then stores the player details
997.                    ElseIf count = 2 Then
998.                        'If first player has been sub more than the current player
999.                        If TempTimesHolder > Holder(1, 2) Then
1000.                           Holder(2, 1) = Holder(1, 1)
1001.                           Holder(2, 2) = Holder(1, 2)
1002.                           Holder(1, 1) = TempIDHolder
1003.                           Holder(1, 2) = TempTimesHolder
1004.                           'If current player has been sub more than the first player
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                    Else
                        Holder(2, 1) = TempIDHolder
                        Holder(2, 2) = TempTimesHolder
                    End If
                Else
                    'If the current player has been sub more times than the first holding player
                    If TempTimesHolder > Holder(1, 2) Then
                        'If the current player has been sub more times than the second holding player
                        If TempTimesHolder > Holder(2, 2) Then
                            'Sets second holding player to sub
                            MALPlayerList(Holder(2, 1)).setPosition("SUB")
                            'Shifts the players details in the holder array
                            Holder(2, 1) = Holder(1, 1)
                            Holder(2, 2) = Holder(1, 2)
                            Holder(1, 1) = TempIDHolder
                            Holder(1, 2) = TempTimesHolder
                        Else
                            'Sets second holding player to sub
                            MALPlayerList(Holder(2, 1)).setPosition("SUB")
                            'adds the players details in the holder array
                            Holder(2, 1) = TempIDHolder
                            Holder(2, 2) = TempTimesHolder
                        End If
                    Else
                        'Sets player to sub
                        MALPlayerList(Options(count)).setPosition("SUB")
                    End If
                End If
            Next
            'Sets the players who aren't sub their position
            MALPlayerList(Holder(1, 1)).SetPosition(Positions(positionindex))
            MALPlayerList(Holder(2, 1)).SetPosition(Positions(positionindex))
        End Sub
        Private Sub BTN_SaveTeam_Click(sender As Object, e As EventArgs) Handles BTN_SaveTeam.Click
            Dim ATeam As ArrayList
            Dim ASubs As ArrayList
            Dim NumberOfAPlayingPreferred As Integer
            Dim NumberOfBPlayingPreferred As Integer
            ATeam = TeamA.GetTeam
            'Works out number of players playing preferred position
            'Saves players position and team into database
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1046.                For Count = 0 To ATeam.Count - 1
1047.                    SavePlayersTeamDetails(ATeam(Count).GetIDNumber, "A", ATeam(Count).GetPosition, MALSessionsList(IndexValue).GetIDN
     umber)
1048.                    If ATeam(Count).GetPosition = ATeam(Count).GetFavouritePosition Then NumberOfAPlayingPreferred += 1
1049.                Next
1050.                ASubs = TeamA.GetSubs
1051.                'Saves players subs position into the database
1052.                For Count = 0 To ASubs.Count - 1
1053.                    SavePlayersTeamDetails(ASubs(Count).GetIDNumber, "A", ASubs(Count).GetPosition, MALSessionsList(IndexValue).GetIDN
     umber)
1054.                Next
1055.                Dim BTeam As ArrayList
1056.                Dim BSubs As ArrayList
1057.                BTeam = TeamB.GetTeam
1058.                'Works out number of players playing preferred position
1059.                'Saves players position and team into database
1060.                For Count = 0 To BTeam.Count - 1
1061.                    SavePlayersTeamDetails(BTeam(Count).GetIDNumber, "B", BTeam(Count).GetPosition, MALSessionsList(IndexValue).GetIDN
     umber)
1062.                    If BTeam(Count).GetPosition = BTeam(Count).GetFavouritePosition Then NumberOfBPlayingPreferred += 1
1063.                Next
1064.                'Saves players subs position into the database
1065.                BSubs = TeamB.GetSubs
1066.                For Count = 0 To BSubs.Count - 1
1067.                    SavePlayersTeamDetails(BSubs(Count).GetIDNumber, "B", BSubs(Count).GetPosition, MALSessionsList(IndexValue).GetIDN
     umber)
1068.                Next
1069.                'Saves floating centre details into database
1070.                If FloatingCentre <> -1 Then
1071.                    SavePlayersTeamDetails(MALPlayerList(FloatingCentre).GetIDNumber, "F", "C", MALSessionsList(IndexValue).GetIDNumbe
     r)
1072.                End If
1073.                'Updates stats into the labels
1074.                AStats.Show()
1075.                BStats.Show()
1076.                'Displays the stats
1077.                AStats.Text = AStats.Text & Environment.NewLine & "MVP Points - " & TeamA.GetMVPPoints
1078.                AStats.Text = AStats.Text & Environment.NewLine & "Number playing position" & NumberOfAPlayingPreferred
1079.                BStats.Text = BStats.Text & Environment.NewLine & "MVP Points - " & TeamB.GetMVPPoints
1080.                BStats.Text = BStats.Text & Environment.NewLine & "Number playing position" & NumberOfBPlayingPreferred
1081.                BTN_SaveTeam.Hide()
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1082.               BTN_GenerateTeams.Hide()
1083.               BTN_SendTeam.Show()
1084.           End Sub
1085.           Private Sub BTN_Cancel_Click(sender As Object, e As EventArgs) Handles BTN_Cancel.Click
1086.               Me.Close()
1087.           End Sub
1088.           Private Sub BTN_SendTeam_Click(sender As Object, e As EventArgs) Handles BTN_SendTeam.Click
1089.               Dim EmailBody As String
1090.               Dim ATeam, BTeam, ASubs, BSubs As ArrayList
1091.               'Sends the team details into email
1092.               'Creates email body with each teams
1093.               EmailBody = "Hi All,"
1094.               EmailBody = EmailBody & Environment.NewLine & "The Teams for " & MALSessionsList(IndexValue).GetSessionDate & " have b
    een generated."
1095.               ATeam = TeamA.GetTeam
1096.               BTeam = TeamB.GetTeam
1097.               EmailBody = EmailBody & Environment.NewLine & "Team A"
1098.               For Count = 0 To ATeam.Count - 1
1099.                   EmailBody = EmailBody & Environment.NewLine & ATeam(Count).GetPosition & "  -  " & ATeam(Count).GetFullName
1100.               Next
1101.               ASubs = TeamA.GetSubs
1102.               For Count = 0 To ASubs.Count - 1
1103.                   EmailBody = EmailBody & Environment.NewLine & "Subs" & "  -  " & ASubs(Count).GetFullName
1104.               Next
1105.               EmailBody = EmailBody & Environment.NewLine & "Team B"
1106.               For Count = 0 To ATeam.Count - 1
1107.                   EmailBody = EmailBody & Environment.NewLine & BTeam(Count).GetPosition & "  -  " & BTeam(Count).GetFullName
1108.               Next
1109.               BSubs = TeamB.GetSubs
1110.               For Count = 0 To BSubs.Count - 1
1111.                   EmailBody = EmailBody & Environment.NewLine & "Subs" & "  -  " & BSubs(Count).GetFullName
1112.               Next
1113.               If FloatingCentre <> -1 Then
1114.                   EmailBody = EmailBody & Environment.NewLine & "Floating Centre" & "  -  " & MALPlayerList(FloatingCentre).GetFullN
    ame
1115.               End If
1116.               EmailBody = EmailBody & Environment.NewLine & "See you all there,"
1117.               EmailBody = EmailBody & Environment.NewLine & "Rodborough Rockets"
1118.               Try
1119.                   Dim Smtp_Server As New SmtpClient
1120.                   Dim e_mail As New MailMessage()
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1121.                 Smtp_Server.UseDefaultCredentials = False
1122.                 Smtp_Server.Credentials = New Net.NetworkCredential("rodboroughrocketclub@gmail.com", "CompSciNea2020")
1123.                 Smtp_Server.Port = 587
1124.                 Smtp_Server.EnableSsl = True
1125.                 Smtp_Server.Host = "smtp.gmail.com"
1126.                 e_mail = New MailMessage()
1127.                 e_mail.From = New MailAddress("rodboroughrocketclub@gmail.com")
1128.                 'Finds email address for each player
1129.                 For Count = 0 To MALPlayerList.Count - 1
1130.                     e_mail.Bcc.Add(MALPlayerList(Count).GetEmailAddress())
1131.                 Next
1132.                 'Sends email
1133.                 e_mail.Subject = "Rodborough Rockets Team Generated"
1134.                 e_mail.IsBodyHtml = False
1135.                 e_mail.Body = EmailBody
1136.                 Smtp_Server.Send(e_mail)
1137.                 MsgBox("Mail Sent")
1138.             Catch error_t As Exception
1139.                 MsgBox(error_t.ToString)
1140.             End Try
1141.             Me.Close()
1142.         End Sub
1143.     End Class
1144.     Class Team
1145.         Protected PlayerList As New ArrayList
1146.         Protected SubsList As New ArrayList
1147.         Protected MVPPointsTotal As Integer
1148.         Protected NoPlayingPreferredPosition As Integer
1149.         Public Sub New()
1150.             'Creates new team
1151.             MVPPointsTotal = 0
1152.             NoPlayingPreferredPosition = 0
1153.         End Sub
1154.         Public Sub AddPlayerToTeam(ByVal Player As Player, ByVal MVPPoints As Integer, ByVal PreferredPosition As Boolean)
1155.             'Adds player to the team
1156.             PlayerList.Add(Player)
1157.             MVPPointsTotal = MVPPointsTotal + MVPPoints
1158.             If PreferredPosition Then
1159.                 Me.NoPlayingPreferredPosition = Me.NoPlayingPreferredPosition + 1
1160.             End If
1161.         End Sub
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
1162.          Public Sub AddPlayerToSub(ByVal Player As Player, ByVal MVPPoints As Integer)
1163.              'Add sub to the team
1164.              SubsList.Add(Player)
1165.              MVPPointsTotal = MVPPointsTotal + MVPPoints
1166.          End Sub
1167.          Public Function GetMVPPoints() As Integer
1168.              'Return MVP Points
1169.              Return MVPPointsTotal
1170.          End Function
1171.          Public Function GetTeam() As ArrayList
1172.              'Return Team
1173.              Return PlayerList
1174.          End Function
1175.          Public Function GetSubs() As ArrayList
1176.              'Return subs list
1177.              Return SubsList
1178.          End Function
1179.          Public Function NumberOfSums() As Integer
1180.              'Return number of subs
1181.              Return SubsList.Count
1182.          End Function
1183.          Public Function NumberPlayingPreferredPosition() As Integer
1184.              'Returns the number of players playing preferred position
1185.              Return Me.NoPlayingPreferredPosition
1186.          End Function
1187.      End Class
```

WEB FORM

```
1.  Imports System.Data
2.  Imports System.Data.SqlClient
3.
4.  Public Class WebForm1
5.      Inherits System.Web.UI.Page
6.      Dim AvailableToPlay As String
7.      Dim sConnectionString As String = "server=127.0.0.1;user=root;database=rodborough_rockets;port=3306;password=;"
8.      Dim myConnection As New MySql.Data.MySqlClient.MySqlConnection(sConnectionString)
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
9.      Dim TodaysDate As Date = Date.Today
10.     Public Shared SessionsList As New ArrayList
11.     Public Shared PlayerList As New ArrayList
12.     Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
13.         'Loads the page
14.         'First time loading the page
15.         If Page.IsPostBack = False Then
16.             'Reads the players from the database
17.             ReadPlayers()
18.             'Adds players to the select box
19.             For Count = 0 To PlayerList.Count - 1
20.                 PlayerSelector.Items.Add(PlayerList(Count).GetDisplayValue)
21.             Next
22.             'Reads the sessions from the database
23.             ReadSessions()
24.             'Adds to the sessions to the select box
25.             For Count = 0 To SessionsList.Count - 1
26.                 SessionSelector.Items.Add(SessionsList(Count).GetDisplayValue)
27.             Next
28.         End If
29.     End Sub
30.
31.     Function ConvertDateToSQLDate(ByVal NonSQLDate As Date) As String 'Converts Date to SQL form
32.         Dim DateINSQLForm As String
33.         'Converts date into the SQL form
34.         DateINSQLForm = NonSQLDate.Year & "-" & NonSQLDate.Month & "-" & NonSQLDate.Day
35.         'Return the SQL Form
36.         Return DateINSQLForm
37.     End Function
38.     Sub ReadSessions()
39.         Dim msSqlDate As String
40.         Dim dr As MySql.Data.MySqlClient.MySqlDataReader
41.         Dim conn As New MySql.Data.MySqlClient.MySqlConnection
42.         Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
43.         conn.ConnectionString = sConnectionString
44.         'Converts date to the correct sql format
45.         msSqlDate = ConvertDateToSQLDate(TodaysDate)
46.         'Error catching
47.         Try
48.             'Opens connection to database
49.             conn.Open()
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
            cmd.Connection = conn
            'Reads session from database
            cmd.CommandText = ("SELECT * FROM `session` WHERE `Date` > @Date ORDER by `Date` ASC")
            cmd.Prepare()
            'Sets parameters value
            cmd.Parameters.AddWithValue("@Date", msSqlDate)
            'Reads from the database
            dr = cmd.ExecuteReader()
            While dr.Read
                'Adds new instance to the session
                Dim NewSession As New Sessions(dr("SessionID"), dr("Location"), dr("Date"), dr("Start_Time").ToString, dr("End_Time")
.ToString)
                SessionsList.Add(NewSession)
            End While
            'Closes connection to the database
            conn.Close()
        Catch ex As Exception
            'Displays error message
            MsgBox(ex.ToString)
        End Try
    End Sub
    Sub ReadPlayers()
        Dim conn As New MySql.Data.MySqlClient.MySqlConnection
        Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
        Dim dr As MySql.Data.MySqlClient.MySqlDataReader
        conn.ConnectionString = sConnectionString
        Try
            'Opens connection between database
            conn.Open()
            cmd.Connection = conn
            'Sets command text to read players from database
            cmd.CommandText = "SELECT * FROM `players`;"
            cmd.Prepare()
            dr = cmd.ExecuteReader()
            While dr.Read
                'Creates new instance of player
                Dim NewPlayer As New Players(CInt(dr("ID_Number")), dr("First_Name"), dr("Surname"), dr("Date_Of_Birth"))
                'Adds new player to the Array List
                PlayerList.Add(NewPlayer)
            End While
            'Closes connection to the database
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
            conn.Close()
        Catch ex As Exception
            'Displays error message
            MsgBox(ex.ToString)
        End Try
    End Sub
    'When OK Button Clicked
    Protected Sub BTN_OK_Click(sender As Object, e As EventArgs) Handles BTN_OK.Click
        'If they haven't selected a valid input
        If Yes.Checked And No.Checked Or (Yes.Checked = False And No.Checked = False) Then
                    MsgBox("Error - Must Select whether available to play")
                    AvailableToPlay = "E"
                    'Selected they can attend
                ElseIf Yes.Checked Then
                    AvailableToPlay = "Y"
                    'Adds to database
                    AddToDatabase()
                ElseIf No.Checked Then
                    AvailableToPlay = "N"
                    'Adds to database
                    AddToDatabase()
                End If
            End Sub
            Function CheckWhetherGivenAttendance(ByVal SessionIndex As Integer, ByVal Playerindex As Integer)
                Dim readvalue As Integer = 0
                Dim msSqlDate As String
                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
                Dim dr As MySql.Data.MySqlClient.MySqlDataReader
                conn.ConnectionString = sConnectionString
                Try

                    'Opens database
                    conn.Open()
                    cmd.Connection = conn
                    'Converts todays date to sql form
                    msSqlDate = ConvertDateToSQLDate(TodaysDate)
                    'Checks if they are already given their attendance
                    cmd.CommandText = ("SELECT Count(*) From `attendance` Where `attendance`.`SessionID` = @SessionID and `attendance`
    .`PlayerID` = @PlayerID;")
                    cmd.Prepare()
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
130.                    'Sets parameters value
131.                    cmd.Parameters.AddWithValue("@PlayerID", PlayerList(Playerindex).GetIDNumber)
132.                    cmd.Parameters.AddWithValue("@SessionID", SessionsList(SessionIndex).GetIDNumber)
133.                    dr = cmd.ExecuteReader()
134.                    While dr.Read
135.                        'Sets read value to the number of entries in the database
136.                        readvalue = CInt(dr("Count(*)"))
137.                    End While
138.                    'Closes database
139.                    conn.Close()
140.                    'If the read value is greater than 0 return false
141.                    If readvalue >= "1" Then
142.                        MsgBox(True)
143.                    Else
144.                        'If read value is less then returns true
145.                        MsgBox(False)
146.                    End If
147.                Catch ex As Exception
148.                    'Displays error message
149.                    MsgBox(ex.ToString)
150.                End Try
151.                Return False
152.            End Function
153.            Sub RemoveAttendance(ByVal playerindex As Integer, ByVal Sessionindex As Integer)
154.                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
155.                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
156.                conn.ConnectionString = sConnectionString
157.                Try
158.                    'Opens database
159.                    conn.Open()
160.                    cmd.Connection = conn
161.                    'Sets command text to delete from the attendance if they can't attend
162.                    cmd.CommandText = "DELETE FROM `attendance` WHERE `SESSIONID` = @SessionID and `PlayerID` = @PlayerID;"
163.                    cmd.Prepare()
164.                    'Sets the parameters
165.                    cmd.Parameters.AddWithValue("@SessionID", SessionsList(Sessionindex).GetIDNumber)
166.                    cmd.Parameters.AddWithValue("@PlayerID", PlayerList(playerindex).GetIDNumber)
167.                    cmd.ExecuteNonQuery()
168.                    'Closes database
169.                    conn.Close()
170.                Catch ex As Exception
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
171.                 'Displays error message
172.                 MsgBox("Error")
173.             End Try
174.         End Sub
175.         Sub AddAttendance(ByVal playerindex As Integer, ByVal SessionIndex As Integer)
176.             Dim conn As New MySql.Data.MySqlClient.MySqlConnection
177.             Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
178.             conn.ConnectionString = sConnectionString
179.             Try
180.                 'Opens connection to database
181.                 conn.Open()
182.                 cmd.Connection = conn
183.                 'Sets commant text to add attendance
184.                 cmd.CommandText = "INSERT INTO `attendance`(`SessionID`, `PlayerID`) VALUES (@SessionID,@PlayerID)"
185.                 cmd.Prepare()
186.                 'Sets parameters value
187.                 cmd.Parameters.AddWithValue("@SessionID", SessionsList(SessionIndex).GetIDNumber)
188.                 cmd.Parameters.AddWithValue("@PlayerID", PlayerList(playerindex).GetIDNumber)
189.                 cmd.ExecuteNonQuery()
190.                 'Closes connection to the database
191.                 conn.Close()
192.             Catch ex As Exception
193.                 'Displays error message
194.                 MsgBox(ex.ToString)
195.             End Try
196.         End Sub
197.         Sub AddToDatabase()
198.             Dim playerindex As Integer
199.             Dim PlayerID As Integer
200.             Dim SessionID As Integer
201.             Dim sessionindex As Integer
202.             Dim DAteOfBirth As Date
203.             Dim AreAnAdult As Boolean
204.             Dim InsuranceStartDate As String
205.             Dim AmountDue As Integer
206.             'Gets index of session selected
207.             'Loops through list of sessions
208.             For Count = 0 To SessionsList.Count - 1
209.                 'Checks if session selected equals the session display value
210.                 If SessionsList(Count).GetDisplayValue = SessionSelector.Text Then
211.                     sessionindex = Count
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                    End If
                Next
                'Gets index of player selected
                'Loops through list of player
                For Count = 0 To PlayerList.Count - 1
                    'Checks if player selected equals the player display value
                    If PlayerList(Count).GetDisplayValue = PlayerSelector.Text Then
                        playerindex = Count
                    End If
                Next
                'If valid input
                If AvailableToPlay <> "E" Then
                    'Selected they can play
                    If AvailableToPlay = "Y" Then
                        'Haven't given attendance
                        If CheckWhetherGivenAttendance(PlayerID, SessionID) = False Then
                            'Adds attendance to the database
                            AddAttendance(playerindex, sessionindex)
                            'Gets player selected date of birth
                            DAteOfBirth = PlayerList(playerindex).GetDOFB
                            'Checks whether the player is an adult
                            AreAnAdult = GetWhetherAdult(DAteOfBirth, SessionsList(sessionindex).GetDAte)
                            'If they are an adult
                            If AreAnAdult = True Then
                                'If the player has insurance
                                If HasInsurance(SessionsList(sessionindex).GetSeason, PlayerList(playerindex).GetIDNumber, InsuranceSt
artDate) Then
                                    'Counts number of attendances the player has made
                                    If CountsNumberOfAttendances(SessionsList(sessionindex).GetDate, SessionsList(sessionindex).GetIDN
umber, SessionsList(sessionindex).GetSeason) < 21 Then
                                        'Amount due = 2
                                        AmountDue = 2
                                    Else
                                        'Amount due =3
                                        AmountDue = 3
                                    End If
                                Else
                                    'Amount due =3
                                    AmountDue = 3
                                End If
                            Else
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                                'Amount due = 1
                                AmountDue = 1
                            End If
                            'Displays amount due
                            MsgBox("Amount due " & AmountDue)
                        End If
                    ElseIf AvailableToPlay = "N" Then
                        'Removes players attendance from the database if they can't attend
                        RemoveAttendance(playerindex, sessionindex)
                    End If
                End If
                'Displays message
                MsgBox("Thank you for entering whether you will be attending the session")
            End Sub
            Function CountsNumberOfAttendances(ByVal mdDate As Date, ByVal msIDNumber As String, ByVal msSeason As String)
                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
                Dim dr As MySql.Data.MySqlClient.MySqlDataReader
                Dim mISelectCount As Integer
                Dim msSQLSessionDate As String
                'Converts date to sql form
                msSQLSessionDate = ConvertDateToSQLDate(mdDate)
                Try
                    conn.ConnectionString = sConnectionString
                    'Opens database
                    conn.Open()
                    cmd.Connection = conn
                    'counts number of time they have attended
                    cmd.CommandText = ("SELECT Count(*) From `attendance` INNER JOIN `session` on `session`.`SessionID` = `attendance`
    .`SessionID` Where `session`.`Date` <= @Date and `PlayerID` = @IDNumber and `session`.`Season`= @Season;")
                    cmd.Prepare()
                    'Sets parameters value
                    cmd.Parameters.AddWithValue("@Date", mdDate)
                    cmd.Parameters.AddWithValue("@IDNumber", msIDNumber)
                    cmd.Parameters.AddWithValue("@Season", msSeason)
                    dr = cmd.ExecuteReader()
                    While dr.Read
                        'Stores number of entries
                        mISelectCount = dr("COUNT(*)")
                    End While
                    'Closes database
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
                    conn.Close()
                Catch ex As Exception
                    MsgBox(ex.ToString)
                End Try
                'Returns number of session attended
                Return mISelectCount
            End Function
            Function GetWhetherAdult(ByVal Input As Date, ByVal DateToCompare As Date)
                Dim DOFBChecker As Date
                DOFBChecker = Input
                DOFBChecker = DOFBChecker.AddYears(18) 'Adds 18 years on to selected date
                If DateToCompare.ToShortDateString >= DOFBChecker Then 'Checks whether the todays date is greater than the Date select
    ed plus 18 years to workout whether they are an adult
                    Return True 'If they are an adult returns true
                Else
                    Return False 'If They aren't an adult returns false
                End If
            End Function
            Function HasInsurance(ByVal msSeason As String, ByVal msIDNumber As String, ByRef msInsuranceStartDate As String) As Boole
    an
                Dim conn As New MySql.Data.MySqlClient.MySqlConnection
                Dim cmd As New MySql.Data.MySqlClient.MySqlCommand
                Dim dr As MySql.Data.MySqlClient.MySqlDataReader
                Dim mISelectCount As Integer
                conn.ConnectionString = sConnectionString
                Try
                    'Opens connection with database
                    conn.Open()
                    cmd.Connection = conn
                    'Reads whether they have insurance for the season and the start date
                    cmd.CommandText = ("SELECT COUNT(*) , `Start_Date` From `insurance` Where `Season` = @Season and `ID_Number` = @ID
    Number;")
                    cmd.Prepare()
                    'Sets parameters value
                    cmd.Parameters.AddWithValue("@Season", msSeason)
                    cmd.Parameters.AddWithValue("@IDNumber", msIDNumber)
                    'Reads from database
                    dr = cmd.ExecuteReader()
                    While dr.Read
                        'Stores values
                        mISelectCount = dr("COUNT(*)").ToString
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
329.                     msInsuranceStartDate = dr("Start_Date").ToString
330.                 End While
331.                 'Closes connection to the database
332.                 conn.Close()
333.             Catch ex As Exception
334.                 MsgBox(ex.ToString)
335.             End Try
336.             'Returns whether they have insurance
337.             If mISelectCount = 1 Then
338.                 Return True
339.             Else
340.                 Return False
341.             End If
342.         End Function
343.     End Class
344.     Class Players
345.         Protected FirstName As String
346.         Protected Surname As String
347.         Protected IDNumber As Integer
348.         Protected DateOfBirth As Date
349.         'Constructor class
350.         Public Sub New(ByVal IDNumber As Integer, ByVal FirstName As String, ByVal Surname As String, ByVal DofB As Date)
351.             Me.FirstName = FirstName
352.             Me.Surname = Surname
353.             Me.IDNumber = IDNumber
354.             Me.DateOfBirth = DofB
355.         End Sub
356.         'Returns display value
357.         Public Function GetDisplayValue() As String
358.             Return (Me.IDNumber & " - " & Me.FirstName & " " & Me.Surname)
359.         End Function
360.         'Returns ID Number
361.         Public Function GetIDNumber() As String
362.             Return Me.IDNumber
363.         End Function
364.         'Returns Date of Birth
365.         Public Function GetDOFB() As String
366.             Return Me.DateOfBirth
367.         End Function
368.     End Class
369.     Class Sessions
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

```
370.           Protected Location As String
371.           Protected StartTime As String
372.           Protected DateOfSession As Date
373.           Protected IDNumber As Integer
374.           Protected Season As String
375.           'Constructor class
376.           Public Sub New(ByVal IDNumber As Integer, ByVal Location As String, ByVal StartTime As String, ByVal DateOFSession As Date
    , ByVal Season As String)
377.               Me.IDNumber = IDNumber
378.               Me.StartTime = StartTime
379.               Me.Location = Location
380.               Me.DateOfSession = DateOFSession
381.               Me.Season = Season
382.           End Sub
383.           'Returns Display Value
384.           Public Function GetDisplayValue() As String
385.               Return (Me.DateOfSession & " - " & Me.StartTime & " @ " & Me.Location)
386.           End Function
387.           'Returns ID Number Value
388.           Public Function GetIDNumber() As String
389.               Return Me.IDNumber
390.           End Function
391.           'Returns Date of Session
392.           Public Function GetDate() As Date
393.               Return DateOfSession
394.           End Function
395.           'Returns season of the session
396.           Public Function GetSeason() As String
397.               Return Season
398.           End Function
399.       End Class
```

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

ALPHA TESTING

VB FORMS TESTS

MAIN MENU FORM TESTS

Video - https://www.youtube.com/watch?v=zpeWT6gC4cs

Test Number	Purpose	Test Data	Expected Result	Pass / Fail	Cross Reference
1	Tests Add Player Button	Add Player Button Clicked	Opens Add Player Form	Pass	6 seconds
2	Tests Modify Player's Details Button	Modify Player's Details Button Clicked	Opens Modify Player's Details Form	Pass	17 seconds
3	Tests Remove Player Button	Remove Player Button Clicked	Opens Remove Player Form	Pass	22 seconds
4	Tests Player's Attendance Button	Add Attendance Button Clicked	Opens Add Attendance Form	Pass	31 Seconds
5	Tests Session Button	Add Session Button Clicked	Opens Add Session Form	Pass	39 Seconds
6	Tests Award MVP Points Button	Award MVP Points Button Clicked	Opens Award MVP Points Form	Pass	50 Seconds
7	Tests Generate Team Button	Generate Teams Button Clicked	Opens Generate Teams Form	Pass	56 Seconds

ADD PLAYER FORM TESTS

Video - https://youtu.be/QwMeiw5hJ0A

8.1	Validates First Name Input	Owain	Accepts First Name	Pass	2 mins 42 sec
8.2	Validates First Name Input		Rejects First Name	Pass	32 Seconds
9.1	Validates Surname Input	Lansdowne	Accepts Surname	Pass	2 mins 42 sec
9.2	Validates Surname Input		Rejects Surname	Pass	40 Seconds
10.1	Validates Preferred Netball position is a netball position	GK	Accepts Position	Pass	2 mins 42 sec
10.2	Validates Preferred Netball position is a netball position	HG	Rejects Position	Pass	97 seconds
11.1	Validates Second favourite Netball position is a netball position	GA	Accepts Position	Pass	2 mins 42 sec

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

11.2	Validates Second favourite Netball position is a netball position	HG	Rejects Position	Pass	1 mins 47 sec
12.1	Validates third favourite Netball position is a netball position	GS	Accepts Position	Pass	2 mins 42 sec
12.2	Validates third favourite Netball position is a netball position	HG	Rejects Position	Pass	1 mins 57 sec
13.1	Validates three favourite netball positions are different	GK GA GS	Accepts position	Pass	2 mins 42 sec
13.2	Validates three favourite netball positions are different	GK GK GK	Rejects position	Pass	2 mins 06 sec
14.1	Validates email entered is in a valid form	189503@ godalming. ac.uk	Accepts Email Address	Pass	2 mins 42 sec
14.2	Validates email entered is in a valid form	owain. Lansdowne	Rejects Email Address	Pass	1 mins 22 sec
15.1	Checks user has selected whether they have insurance	Yes Clicked	Accepts input	Pass	2 mins 42 sec
15.2	Checks user has selected whether they have insurance	Nothing selected	Rejects input	Pass	24 secs
16.1	Checks user has selected whether they have insurance for next season	Yes Clicked	Accepts input	Pass	3 mins 47 secs
16.2	Checks user has selected whether they have insurance for next season	Nothing selected	Rejects input	Pass	
17	Reads highest ID Number and assigns one above	Currently highest is 20	Assigns Player ID Number 21 and displays it	Pass	2 mins 38 sec
18	Saves player's details to database	OK Button clicked	Saved details into the database	Pass	2 mins 51 sec
19	Saves player's insurance details to the database	OK Button Clicked	Saved details into the database	Pass	2 mins 58 sec
20	Saves player's favourite positions to the database	OK Button Clicked	Saved details into the database	Pass	3 mins 14 sec
21	Saves player's next season insurance details to the database	OK Button Clicked	Saved details into the database	Pass	4 mins 03 sec
22	Cancel button clicked form closes	Cancel Button Clicked	Form Closes	Pass	4 mins 05 sec

MODIFY PLAYER DETAILS FORM TEST

Video - https://www.youtube.com/watch?v=pV8zcHjEfvM

23	Reads the list of player's and details from the database	Modify Player Details button clicked	Displays list of players in list box	Pass	12 secs
24.1	Validates player selected is valid	23 – Player Edited	Accepts player and displays their current details	Pass	35 Secs
24.2	Validates player selected is valid	Player	Rejects player and displays error message	Pass	26 secs
25.1	Validates First Name Input	Player	Accepts First Name	Pass	2 mins 57 sec
25.2	Validates First Name Input		Rejects First Name	Pass	1 min 11 sec
26.1	Validates Surname Input	Has been Edited	Accepts Surname	Pass	2 mins 57 sec
26.2	Validates Surname Input		Rejects Surname	Pass	1 min 20 sec
27.1	Validates Preferred Netball position is a netball position	GK	Accepts Position	Pass	2 mins 57 sec
27.2	Validates Preferred Netball position is a netball position	HA	Rejects Position	Pass	2 mins 05 sec
28.1	Validates Second favourite Netball position is a netball position	GA	Accepts Position	Pass	2 mins 57 sec
28.2	Validates Second favourite Netball position is a netball position	Ha	Rejects Position	Pass	2 mins 16 sec
29.1	Validates third favourite Netball position is a netball position	GS	Accepts Position	Pass	2 mins 57 sec
29.2	Validates third favourite Netball position is a netball position	Ha	Rejects Position	Pass	2 mins 25 sec
30.1	Validates the three favourite netball positions are different	WD C WA	Accepts position	Pass	2 mins 57 sec
30.2	Validates the three favourite netball positions are different	C C C	Rejects position	Pass	2 mins 35 sec
31.1	Validates email entered is in a valid form	189503@ godalming. ac.uk	Accepts Email Address	Pass	2 mins 57 sec
31.2	Validates email entered is in a valid form	player.netball	Rejects Email Address	Pass	1 min 46 sec

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

32.1	Checks user has selected whether they have insurance	Yes Clicked	Accepts input	Pass	2 mins 57 sec
32.2	Checks user has selected whether they have insurance	Nothing selected	Rejects input	Pass	36 sec
33.1	Checks user has selected whether they have insurance for next season	Yes Clicked	Accepts input	Pass	3 mins 42 secs
33.2	Checks user has selected whether they have insurance for next season	Nothing selected	Rejects input	Pass	
34	Modifies a player's details in the database	OK Button Clicked	Modifies details	Pass	3 mins 03 sec
35	Adds player's insurance details to the database	OK Button Clicked	Saves details into the database	Pass	3 mins 09 sec
36	Adds player's next season insurance details to the database	OK Button Clicked	Saves details into the database	Pass	3 mins 53 sec

ADD Attendance Form Tests

Video - https://www.youtube.com/watch?v=UyJNX3S4uss

37	Reads list of Players from the database	Add Player's Attendance Button Clicked	Adds players to the select box	Pass	6 secs
38	Reads list of Sessions from the database	Add Player's Attendance Button Clicked	Adds session to the select box	Pass	16 secs
39.1	Validates the session the user has selected	07/03/2020 – 20:00 @ Rodborough	Accepts input	Pass	1 mins 32 secs
39.2	Validates the session the user has selected	Next Session	Rejects input	Pass	57 secs
40.1	Validates the player the user has selected	1-Owain Lansdowne	Accepts input	Pass	1 mins 32 secs
40.2	Validates the player the user has selected	Best Player	Rejects input	Pass	45 secs
41.1	Validates the user input of whether they will be attending the session	Yes selected	Accepts input	Pass	1 mins 32 secs
41.2	Validates the user input of whether they will be attending the session	Neither Selected	Rejects input	Pass	32 seconds

42	Calculates amount due	The player selected is an adult, has insurance and has attended less than 21 sessions	Displays message saying " Amount due = 2"	Pass	1 mins 32 sec
43	Saves player's attendance to the database	They will be attending and OK Button Clicked	Saves to the database	Pass	1 mins 57 sec
44	Removes attendance if they can't attend the session	Can't attend and previously said they could.	Removes instance of them attending from the database	Pass	2 mins 25 sec
45	Forms closes when Cancel button clicked	Cancel Button Clicked	Closes form	Pass	2 mins 02 secs

ADD SESSION FORM TESTS

Video - https://www.youtube.com/watch?v=j6v2HWxxNQw

46.1	Validates user's start time input	Start Time = 15:00	Accepts input	Pass	1 mins 37 secs
46.2	Validate's users start time input	Hour : Minute	Rejects input and displays error message	Pass	40 secs
47.1	Validates user's end time input	End Time = 16:00	Accepts input	Pass	1 mins 37 secs
47.2	Validates user's end time input	Hour : Minute	Rejects input and displays error message	Pass	50 secs
48.1	Validates start time is before end time	Start Time = 15:00 End Time = 16:00	Accepts input	Pass	1 mins 37 secs
48.2	Validates start time is before end time	Start Time = 17:00 End Time = 16:00	Rejects input	Pass	1 mins 05 secs
49.1	Validates that the user has entered a Location	Rodborough	Accepts input	Pass	1 mins 37 secs
49.2	Validates that the user has entered a Location		Rejects input and displays error message	Pass	1 mins 21 secs
50	Reads the highest ID Number for sessions in the database and adds 1 for the new session's ID Number	OK Button Clicked Current highest 6	Displays session ID Number	Pass	1 mins 37 secs
51	Saves new session to the database	OK Button Clicked	Adds session to the database	Pass	1 mins 38 secs

52	Sends email out with details about the session	Send Reminder Button Clicked	Email sent	Pass	2 mins 20 secs

REMOVE PLAYER FORM TEST

Video - https://www.youtube.com/watch?v=vhsgycYlJw4

53	Successfully reads list of players from the database	Remove Player Button Clicked and form loaded	Displays players in a list box	Pass	6 seconds
54.1	Validates player to be removed is a player on the list	18 – Player To Be Removed	Accepts player	Pass	30 seconds
54.2	Validates player to be removed is a player on the list	Goalkeeper	Rejects selected player and displays error message	Pass	20 seconds
55	Removes player's personal details from database	OK Button Clicked	Removes details from database	Pass	42 Seconds
56	Removes player's favourite position details from database	OK Button Clicked	Removes details from database	Pass	44 Seconds

AWARD MVP POINTS FORM TESTS

Video - https://youtu.be/T-Z5967XQLs

57	Reads list of sessions from the database	Award MVP Points Button clicked, and form opened	Displays session in select box	Pass	8 Seconds
58.1	Validates session selected is an actual session	25/02/2020 – 20:00 @ Broadwater	Accepts Value	Pass	38 Seconds
58.2	Validates session selected is an actual session	Session 1	Rejects value and displays error message	Pass	29 Seconds
59	Reads list of players who attended the session	Valid session selected	Displays players in a list box	Pass	46 Seconds
60.1	Validates that a valid player has been selected	Owain Lansdowne	Accepts Input	Pass	1 minute 30 seconds
60.2	Validates that a valid player has been selected	Test Player	Rejects Input	Pass	1 minute 15 seconds
61.1	Validates that the same player hasn't been selected 3 times	Owain Lansdowne	Accepts input	Pass	1 minute 30 seconds

		Aaron Ramsey Hadleigh Parkes			
61.2	Validates that the same player hasn't been selected 3 times	Owain Lansdowne Owain Lansdowne Owain Lansdowne	Rejects input and displays error message	Pass	1 minute
62	Saves the best player of the session into the database	OK Button Clicked	Saves data into the database	Pass	1 minute 35 seconds
63	Form closes when Cancel Button Clicked	Cancel Button Clicked	Form Closes	Pass	1 minute 50 seconds

GENERATE TEAMS FORM TESTS

Video - https://www.youtube.com/watch?v=-77XtsTvNdc

White Box Testing Evidence - https://www.youtube.com/watch?v=CVPwG_z5nzc

64	Reads list of sessions from the database	Form Opened	Displays session in a list box	Pass	7 seconds
65.1	Validates that a valid a session has been selected	06/03/2020 - 20:00 @ Rodborough	Accept input	Pass	35 Seconds
65.2	Validates that a valid a session has been selected	Training Session	Rejects input and displays error message	Pass	30 seconds
66	Reads list of players from database	Generate Teams Button Clicked	Teams Generated with player who can attend the session from the database	Pass	35 Seconds
67	Teams generated with 14 players playing	Generate Teams Button Clicked	Teams are displayed with even statistics	Pass	35 Seconds
68	Saves teams details into the database	Save Teams Buttons Clicked	Teams stored into database	Pass	Figure 9 - Data saved in database
69	Email sent with the teams' details in	Send Team Button clicked	Teams sent in an email	Pass	1 minute 44 seconds

70	Form closes when Cancel button clicked	Cancel Button Clicked	Form Closes	Pass	2 minute 04 second
71	Fair Teams generated with 14 players playing but with different set of players	Generate Teams Button Clicked Session 13/03/2020 – 20:00 @ Rodborough	Teams displayed in form	Pass	2 minute 32 seconds
72	Fair Teams generated with 16 players playing	Generate Teams Button Clicked Session 15/03/2020 – 20:00 @ Rodborough	Teams displayed in form	Pass	2 minute 53 seconds
73	Fair Teams generated with 15 players playing	Generate Teams Button Clicked Session 14/03/2020 – 20:00 @ Rodborough	Teams displayed in form	Pass	3 minutes 27 seconds
74	Fair Teams generated with 13 players playing	Generate Teams Button Clicked Session 16/03/2020 – 20:00 @ Rodborough	Teams displayed in form	Pass	4 minutes 02 seconds
75	Fair teams generated with 12 players playing	Generate Teams Button Clicked Session 17/03/2020 – 20:00 @ Rodborough	Teams displayed in form	Pass	4 minutes 42 seconds
76	Teams generated with 14 players with different set of players	Generate Teams Button Clicked Session 18/03/2020 – 20:00 @ Rodborough	White Box Testing	Pass	5 minutes 12 seconds
77	Teams generated with 11 players playing the session	Generate Teams Button Clicked Session 19/03/2020 – 20:00 @ Rodborough	Teams displayed in form	Pass	5 minutes 45 seconds

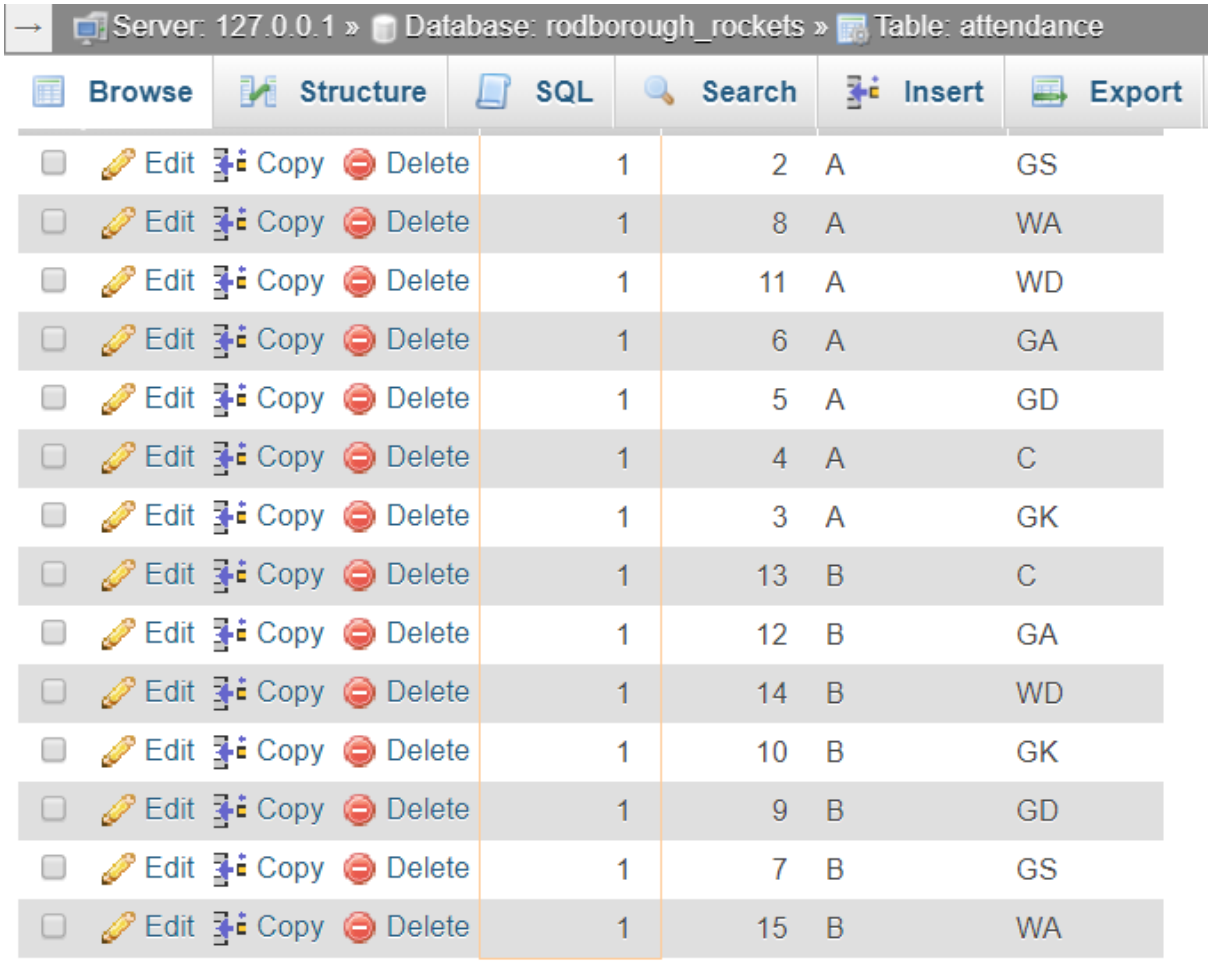


Figure 9 - Data saved in database

WEB FORM TEST

Video - https://www.youtube.com/watch?v=rEqjY-w5i6I

71	Reads player from the database	Web Form Loaded	Add player to the list box	Pass	1 seconds
72	Reads session from the database	Web Form Loaded	Add session to the list box	Pass	18 seconds
73.1	Validates user's input of whether they will be attending the session	Yes Selected	Accepts input	Pass	1 minute 20 seconds
73.2	Validates user's input of whether they will be attending the session	Neither yes or no selected	Displays Error Message	Pass	38 seconds
74	Calculates amount due for the session	Player selected who is an adult, has insurance and has attended less than 21 sessions.	Out puts Amount Due – 2	Pass	1 minute 20 seconds

75	Saves instance of the player attending the session into the database	Yes selected and Ok Button clicked	Saves data to the database	Pass	1 minute 30 seconds
76	Removes instance of the player attending the session into the database	No selected and Ok Button clicked	Removes data from the database	Pass	1 minute 40 seconds

BETA TESTING

Questionnaire completed by Jenny Lansdowne who is Rodborough Rockets Secretary and Player.

1. Are all the requirements met?
   - ☐ Yes
   - ☐ No

   If you selected No, what requirements haven't been met?

   _____
   _____
   _____
   _____

2. Is the User Interface easy to use?
   - ☐ Yes
   - ☐ No

   Any changes you would made to the user interface?
   Date of birth was slightly hard to enter as you had to start with the year and narrow it down.

3. Have you encountered any problems?
   - ☐ Yes
   - ☐ No

   If you selected Yes, what problems have you encountered?

   _____
   _____
   _____
   _____

4. What did you like about the program?
   I like how the MVP points are taken into account when picking the teams to try to ensure that the teams are evenly matched. This seems like a good way to balance the teams.
5. What improvements would you make to the program?
   It would be nice if there was a login system so that players could modify their own details and would only be able to log their own attendance.

EVALUATION

Requirement Number	Description	Requirements met	Comment	Evidence
1 i	Player can add whether they are available online	Yes	User feedback was positive.	Tests 71 – 76
1 ii a	Manager can add a new playe'rs detail	Yes	Met but could tweak the date of birth input to make it easier to use.	Tests 8 -16
1 ii b	Manager can select a player to modify their player Details	Yes	Met in the Modify Player Details form where they can modify each player's details.	Tests 24 – 33
1 ii c	Manager can add enter details for new Session	Yes	Met in the new sessions form where the user can enter the details for the session.	Tests 46 – 49
1 ii d	Manager can select whether the player can attend the session	Yes	Met as the manager can select a session from the list of sessions and select a player and add whether they are planning on attending the session	Tests 39 – 41
1 ii e	Manager can select a player to remove	Yes	Given the list of players and they can select one of the players to remove.	Test 54
1 ii f	Manager can select to generate a team	Yes	Manager is given a list of sessions for which they can generate a team.	Tests 65
1 ii g	Manager can select the 3 best players for the session	Yes	User feedback likes the layout.	Test 60 – 61
2 i	Adds a new player's details to the database	Yes	Fully met as when a new player's details are entered then they are recorded on the database	Test 18
2 ii	Modifies a player's details in the database	Yes	Fully met as the player's details get modified in the database	Test 34
2 iii	Adds a player's attendance	Yes	Fully met and is used in both the main program and the web form	Test 43 & 44
2 iv	Calculates amount due by the player for the session	Yes	Used when the player says they can attend the session on both the web form and the main program.	Test 42

Owain Lansdowne
Candidate Number – 9503
Godalming College
Centre Number - 64395

2 v	Adds new session to the database	Yes	Fully met new session details are saved when session added	Test 51
2 vi	Sends an email out with the details about the new session	Yes	Fully met when a new session is added	Test 52
2 vii	Removes a player from the database when selected	Yes	When a player has been selected then they get removed from the database	Test 55 -56
2 viii	Saves MVP Players for the session into the database	Yes	Fully met as saves three best players from the session and their rank to the database.	Test 62
2 ix	Reads the sessions from the database to select	Yes	Fully met as it reads session so that the player can add whether they are attending the session	Test 64
2 x	Reads the players from the database that are attending the session	Yes	Fully met when the teams are being generated	Test 66
2 xi	Generates even teams	Yes	Fully met when a session has been selected for teams to be generated for.	Test 67 & 71 - 77
2 xii	Saves the teams' details in the database	Yes	Teams are stored when the teams have been generated	Test 68
2 xiii	Sends the team out in the email	Yes	Fully met as once the teams have been generated then the user can select to email them out	Test 69
4 i	New player's ID Number is displayed	Yes	Fully met when a new player is added it displays their new ID Number in a msg box	Test 17
4 ii	Error message displayed if invalid input	Yes	Fully met, when an input is invalid such as not entering a player's first name.	Test 8.2, 9.2, 10.2, 11.2, 12.2, 13.2, 14.2, 15.2, 16.2
4 iii	Email with session details sent	Yes	Fully met, email sent when a new session is entered	Test 52
4 iv	The teams generated will be displayed	Yes	Fully met, when a session has been selected then it displays the teams which have been generated for the session	Test 67

4 v	If the player can attend the session, then it will display the amount the player needs to pay.	Yes	Fully met, displays the amount due when using the main program and the web form.	Test 17
4 vi	The teams will be emailed out.	Yes	Fully met as teams will be emailed out once they have been generated and if the user wishes to email them out	Test 69

I have successfully met all my requirements for the program which can be shown above. It allows the user to add new players, modify a player's details, add a new session, remove a player, award a session's MVP points and generate the teams.
I have spoken to the Rodborough Rockets secretary and, having used the program for a bit, they seem happy with the program as it produces fair even teams which should make the games nice and close at their session. She is also pleased that some of the admin such as calculating how much a player needs to pay for a session is done by the program when the player selects whether they are available for the session. She didn't encounter any problems which using the program.
A feature that the Secretary suggested would be to have a login system for the web form so that only club members could use the form as well as only allowing the player logged in to add their own attendance rather than being able to add another players attendance. In addition she would add the option for the players to modify their own details in the web form so that the manager doesn't have to do this.

If I were to improve my team picker, I would do so by using a more advanced team picker algorithm so that the teams are even closer in ability. This could be done by using more data to help determine the ability of the player rather than just who has played best in the session previously; for example by ranking certain skills such as passing, shooting and defending so that the teams are closer in ability. As well the algorithm could make sure that teams have a range of players with different skills such as having some good passers as well as having some good shooters rather than just having teams of players with similar strengths, as the best teams have a mixture of players will different key strengths.
Another feature I could add is the capability to generate teams for competitive games so that it picks the best team possible for the game with players playing in their best position whilst also ensuring that everyone is given opportunity to play competitive games. The team picker would also need to look at players' attributes to ensure that the team has a mixture of players with strengths as this will mean that the team can play well as a team. It could consider how different partnerships are working so that if the team keeps conceding very few goals then the GK and GD should be kept together and picked again as they are clearly playing well together.