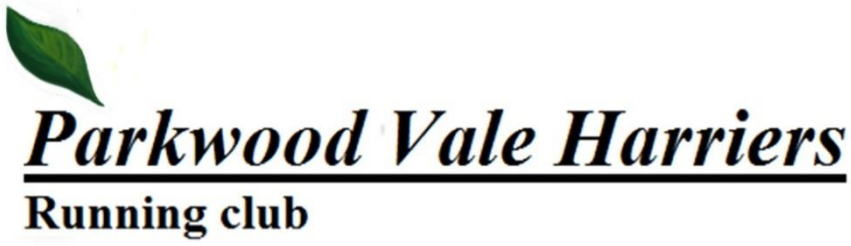


Analysis & Design

Background - Parkwood Vale Harriers

A group of members of the Parkwood Vale Harriers have decided to organise an event to raise money for local charities. They will choose a team of eight people who will run non-stop from John O'Groats to Land's End, in the shortest time possible. They intend to organise the run as a relay with one person running each hour while others rest in the team's minibus. Each runner will run three times per day and will have to be very fit.

The group of members has devised a suggested demanding training programme, which will include running, cycling and swimming.

At the end of the training programme, a team of runners will be selected. The runners want to be able to keep accurate records of their training. The runners intend to use the system to monitor their progress as they train for the run. They want to be able to measure the improvement in their fitness and compare their performance with other members of the team.

The runners have provided you with information about the number of calories burned during one hour of exercise. The running club has commissioned you to create a computer based system which will:

- allow the user to:
  - enter, store, retrieve and amend runners' personal details
  - enter, store and retrieve runners' training information
- allow runners to compare their performance with other runners in each of the three training activities.
- select the team

The program will have additional implementations that will aid in its efficiency. One such implementation will be the introduction of a point system. This point system will allow individual members to earn points by completing specific conditions, e.g. every 500 calories burned will reward a member with 1 point.

These points will be utilized when determining the top 10 members, for participation in the charity run from John O'Groats to Land's End.

This way the top 10 members can simply be chosen by looking at who has the highest number of points earned.

Current Manual System

Currently if members of the Parkwood Vale Harriers club want to keep track of their training progress I will assume it is done largely by hand with a small amount of IT reliance for certain calculations.

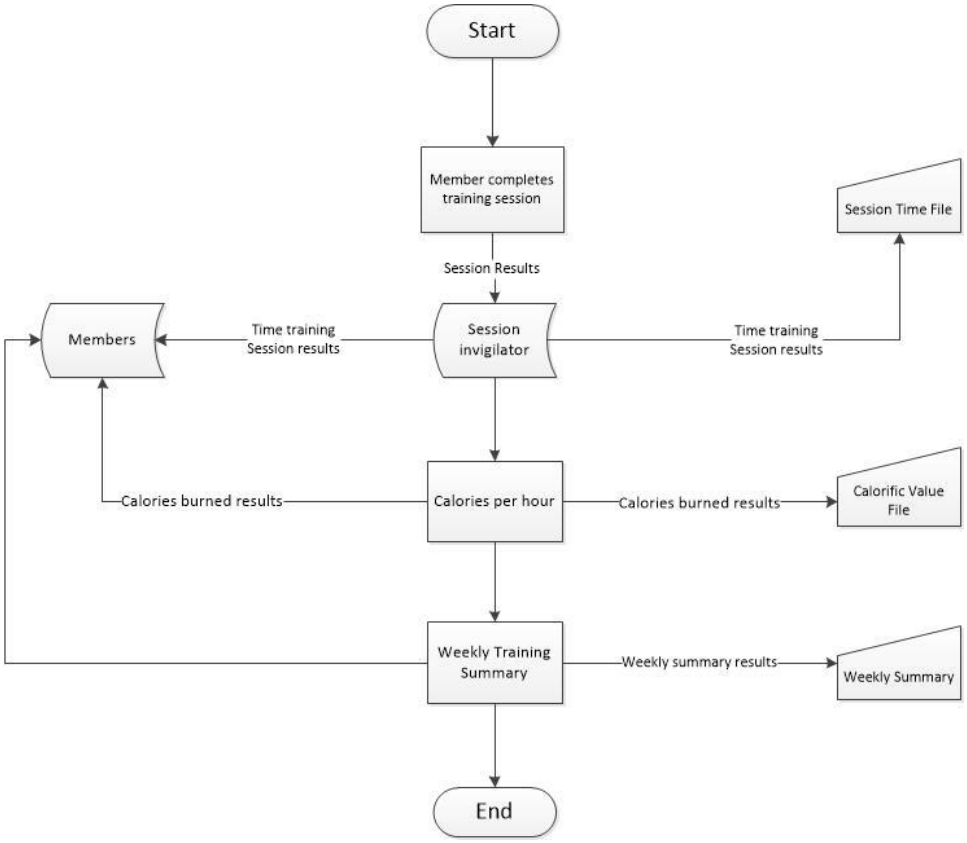
- Member will begin a training session
- Invigilator will begin recording the time duration of the session with a stopwatch
- Once the training session is completed the invigilator will write down the time duration and the distance covered by the member
- Results of the session will be input into a software such as Excel
- Excel will run calculations for session speed
- Excel will also run calculations for calories burned given the results of the training session and the type of activity performed
- Excel will be used to find member's weekly training session summary

If members wish to compare their results and find the top runners for their charity run they would have to do this by

- Sharing each other's training notes
- Adding together their total calories burned in the week
- This number will represent their stamina as it takes into consideration speed and distance as well was the activity

- Members who burn the most calories in the week will be selected for the team of top runners to participate in the charity event

Manual Data Flow Diagram



Aims & Objectives of the System

My aims for the new system are:

- The system will meet all the requirements needed for members of the club to use it effectively, such as the ability to record a training session, update member's details, view calories burned, etc.
- It will be user friendly so that members with little or no IT experience won't have much difficulty using the system effectively
- The system must be as efficient as possible so that member's time is utilised to maximum effect
- The system must be professional so as to reflect the standard and maturity of the Parkwood Vale Harriers running club
- The system must be flexible and allow for many member profiles as is needed for the club's members and allow for an unlimited number of recorded training sessions
- It must allow members to input and edit personal details
- Members must also be able to create and record training sessions
- Members must be able to choose which activity they're recording for each session
- The system must provide instant results on the training sessions
- The system must be able to calculated the number of calories members burn per training session
- The system must be able to calculate members points based on calories
- The system must be able to award bonus points based on the information provided
- The user interface must be self-explanatory which requires very little instructions to use effectively
- The system must be able to give a weekly summary of individual member's training sessions

- The system must be able to select the top 10 members based on the point system
- The system must give a value via a point system which is determined by the amount of activity and the duration of the activity, this will be the value by which the top 10 members can be selected
- Members must be able to compare their training results with other members

Objectives for the new System

- I shall create a range of forms that I will customize which will create a menu driven system. I shall place command buttons on the forms which will enable the end user to easily navigate the system.
- Create a number of forms for data viewing, e.g. FormRank, FormSummary, FormDisplayTraining. These will view via DataGrids and will be set to a read-only preference.
- The data structures embedded within the forms will enable end users to enter training session details and view their details
- Use primary and foreign keys to reduce the duplication of data
- I will create complex algorithms which will automatically calculate: calories burned – based on the activity level and duration, points awarded based on calories, bonus points awarded and selection of the top 10 members based on the points

Limitations of the System

- Every runner can view all the data in the system as there are no access rights. We could add access rights in the future by implementing a sign in system where members have unique username's and passwords.
- The data of the recorded training sessions cannot be changed. This could be improved by having an admin with privileges that include editing previously recorded data via data grids.
- New sport types cannot be added by members as the sports are encoded. However by introducing new lines of code and additional form(s) we would have the ability to add new activities.

Justification of Proposed Solution

The reasons for creating the solution in Visual Basic:

- Through computer programming it is easy to create a system with a huge number of benefits.
- As it is electronic there is no need to do anything by hand, thus less effort to get the same effect.
- Calculations are automatic and can be done very easily.
- All results are processed immediately.
- Microsoft has designed Visual Basic so that it is easy to learn and has a wide range of features that are implemented to make the coding process and overall development of applications a much simpler process.
- Visual Basic does not require additional software for the system to be run, therefore is cheap.
- Visual Basic's Integrated Development Environment (IDE), Visual Studio (2010 specifically) is very efficient and simple to use. It allows me to create a decent user interface using all the elements that are providing, e.g. forms, labels, data grids and buttons. Despite the simplicity and ease of the Visual Studio 2010 is still able to maintain

the professionalism that we are looking for. With the use of these elements I am able to make a self-explanatory and friendly user interface which will allow users to access all the features of the system without needed very much IT experience.

- Visual Basic has a number of data sets which we have used to store data in tables which will contain data on the training sessions and the member details. Using these data structures will allow data to be displayed and edited in grids.
- Visual Basic has various functions that are designed to cut down on coding. There are conversion functions, math functions, string functions, Type and CType functions. All of this will cut down on the time spent coding and will make syntax easier to read.

Data Structures and Methods of Access

The system I have created 3 primary sets of data which I will be storing and accessing data on. The summary data set will store all data on the points and calories. The club data set will store data on the individual the sports, individual members and the training sessions. The results data set will store data on the points and also the members. The method of access we are using is sequential. This essentially means that the items are stored one by one which are accessed one by one in order of how they were stored.

The following tables will outline the data which is stored in each of the sets.

Initial Data Structure

Club Data Set

TableMembers

Field Name	Description	Type	Length	Example Data
ID	This is the number that uniquely identifies the member	Auto-increment Int-32	Variable	1
FirstName	Member's first name	String	Variable	Ryan
SecondName	Member's surname	String	Variable	Jones
DOB	Member's date of birth	DateTime	Variable	10/02/95
Address	Member's address	String	Variable	57 North Walk
City	City the member lives in	String	Variable	Cardiff
Postcode	Post code of member's home residence	Int-32	Variable	CF62 8BX
Gender	Gender of the member	String	Variable	Male
Contact	Contact information of member	Int-32	Variable	07983423039
Email	Member's email address	String	Variable	ryanjones@hotmail.co.uk

TableSports

Field Name	Description	Type	Length	Example Data
ID	A number	Auto-	Variable	2

	that uniquely identifies the activity	Increment Int-32		
Description	A description of the sports activity	String	Variable	Breaststroke

TableTrainingSession

Field Name	Description	Type	Length	Example Data
ID	Unique identifier of the training session	Auto-Increment Int-32	Variable	1
MemberID	This number a unique identifier of the member	Int-32	Variable	3
SportID	A unique identifier for the activity	Int-32	Variable	3
SessionTime	This is the duration of the session	Decimal	Variable	45.5
SessionDistance	The amount of discovered covered over a session	Decimal	Variable	5.5
SessionDate	The date the session was done on	DateTime	Variable	28/01/15
CalorificValue	The amount calories burned for that session	String	Variable	300
Points	Total points awarded for	Decimal	Variable	5

	training session			

DataSets Used for Reports

Summary Data Set

TableByWeek

Field Name	Description	Type	Length	Example Data
Week	Number of each week	Int-32	Variable	7
Calories	This number is the number of calories burned by a member per week	Int-32	Variable	500
Points	This number is the total points acquired by a member per week	Decimal	Variable	2

DataSetResults

TableRunnersPoints

Field Name	Description	Type	Length	Example Data
ID		Int-32	Variable	4
FirstName		String	Variable	Luke
Surname		String	Variable	Jones
Points		Decimal	Variable	2

Entity Relationship Model One

Primary Key
Foreign Key

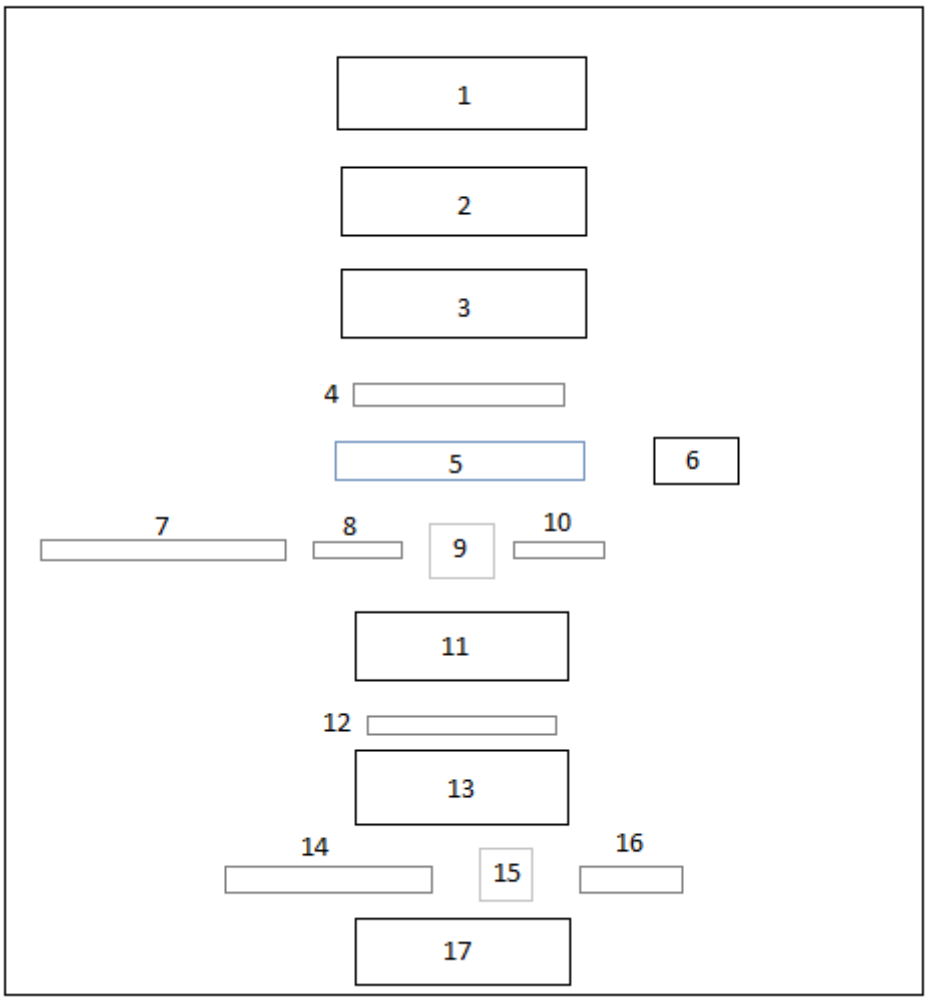
Hardware and Software Requirements

Our system has very basic requirements as the system itself isn't very complex. The minimal requirements for the desktop computers that use this system should be:

Operating System	**Windows 7**
CPU - Processor	1.30GHz
RAM – Memory	6GB
HDD – Hard Drive	500GB
Input-Output	Monitor Mouse Keyboard
Software Requirements:	VB.Net, Microsoft Office (Excel, Word) Windows 7.

User Interface (Forms, Form Properties and Object Names)

TrainingLog - This is the opening form that the user will use to navigate through the entire system.

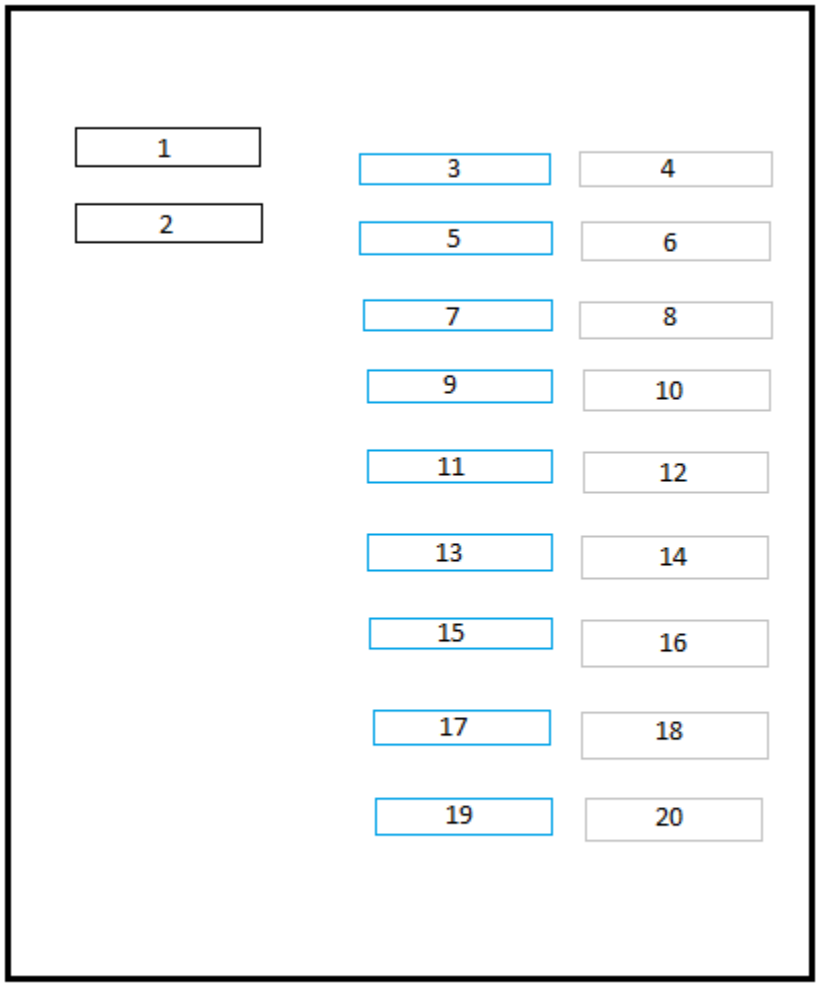


Number	Object	Object Name	Text	Description	Properties
1	Button	ButtonAddNewMember	Add New Member	This button takes the user to the NewRunner form.	Default
2	Button	ButtonViewMemberDetails	View Member Details	This button will take the user to FormMember	Default

				s.	
3	Button	ButtonRecordTrainingSession	Record Training Session	This button will take the user to FormTraining Session	Default
4	Label	Label1	Choose a Member	This text is to describe the purpose of the ComboBoxMembers	Default
5	ComboBox	ComboBoxMembers	N/A	Allows user to choose a member	Default
6	Button	ButtonView	All	Allows the user to select all members rather than chooses individual members via ComboBoxMembers	Default
7	Label	Label7	View a member's training	Describes part of the use of NumericUpDownDays	Default
8	Label	Label5	Last	Describes part of the use of NumericUpDownDays	Default
9	NumericUpDown	NumericUpDownDays	N/A	Allows member to select a number of days in which to view a memer (or all members) training session	Default
10	Label	Label6	Days	Describes part of the use of NumericUpDownDays	Default
11	Button	ButtonDays	Last	This button	Default

			Days	takes the user to FormDisplay Training	
12	Label	Label2	Weekly Summary	Describes the use of ButtonByWeek	Default
13	Button	ButtonByWeek	Summary	This button takes the user to Form_Summary	Default
14	Label	Label3	Rank Runner's Last	Describes the use of the following NumericUpDownRank	Default
15	NumericUpDown	NumericUpDownRank	N/A	Allows member to select a given number of days	Default
16	Label	Label4	Days	Describes the use of NumericUpDownRank	Default
17	Button	ButtonRank	Rank	This button will take the user to FormRank, where they can view the rank of a member within a given number of days, which is selected via the NumericUpDownRank	Default
18	DataSet	DataSetClub	DataSet Club	This dataset will hold all the data on the members, within tables	Default

New Runner – This form will be selected via the Add New Member button in the Training Log, and will allow people to add their personal details and register within the database.

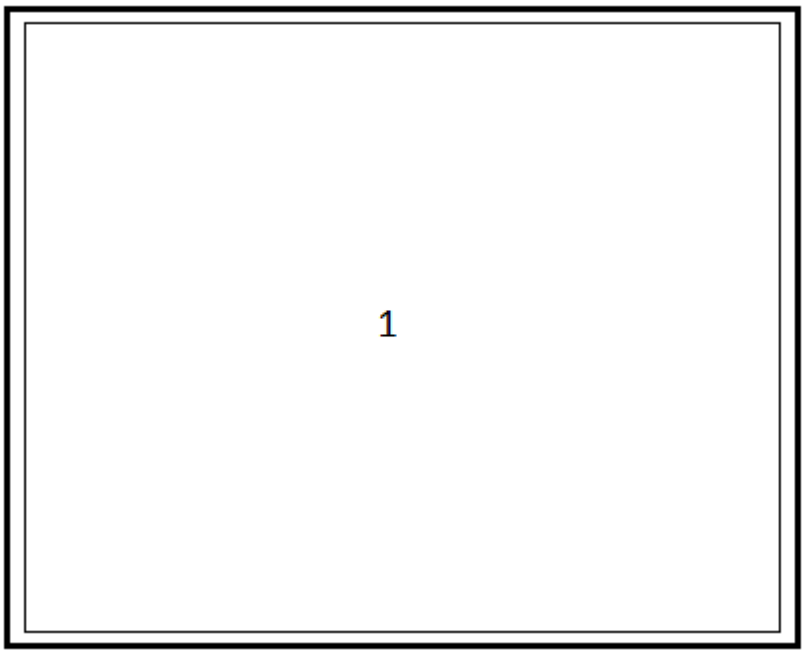


Number	Object	Object Name	Text	Description	Properties
1	Button	ButtonAddNewMember	Add Member	When clicked this button will save all the details held in the TextBoxes and register that member into the database, the form will then close	Default

2	Button	ButtonCancel	Cancel	When clicked this button will reject any data input into the TextBoxes and will close the form	Default
3	Label	LabelFirstName	First Name	Describes what should be input into TextBoxFirstName	Default
4	TextBox	TextBoxFirstName	N/A	This is where user will input their first name	Default
5	Label	LabelSurnameName	Surname	Describes what should be input into TextBoxSurname	Default
6	TextBox	TextBoxSurname	N/A	This is where the user will input their surname	Default
7	Label	LabelDOB	Date of Birth	Describes what the user should put into TextBoxDOB	Default
8	TextBox	TextBoxDOB	N/A	This is where the user should input their date of birth	Default
9	Label	LabelAddress	Address	Describes what the user should input into TextBoxAddress	Default
10	TextBox	TextBoxAddress	N/A	This is where the user will input their address.	Default
11	Label	LabelCity	City	Describes what the user should input into TextBoxCity	Default
12	TextBox	TextBoxCity	N/A	This is where the user will input their current city	Default
13	Label	LabelPostCode	Post Code	This describes what the user should input into TextBoxPostCode	Default
14	TextBox	TextBoxPostCode	N/A	This is where the user will input their Post Code.	Default
15	Label	LabelContact	Contact	Describes what the user should input into LabelTextBox	Default
16	TextBox	TextBoxContact	N/A	This is where the user will input their contact number	Default
17	Label	LabelGender	Gender	Describes what the user should input into TextBoxGender	Default

18	TextBox	TextBoxGender	N/A	This is where the user will input their gender	Default
19	Label	LableEmailAddress	Email Address	Describes what the user should input into TextBoxEmailAddress	Default
20	TextBox	TextBoxEmailAddress	N/A	This is where the user should input their email address.	Default

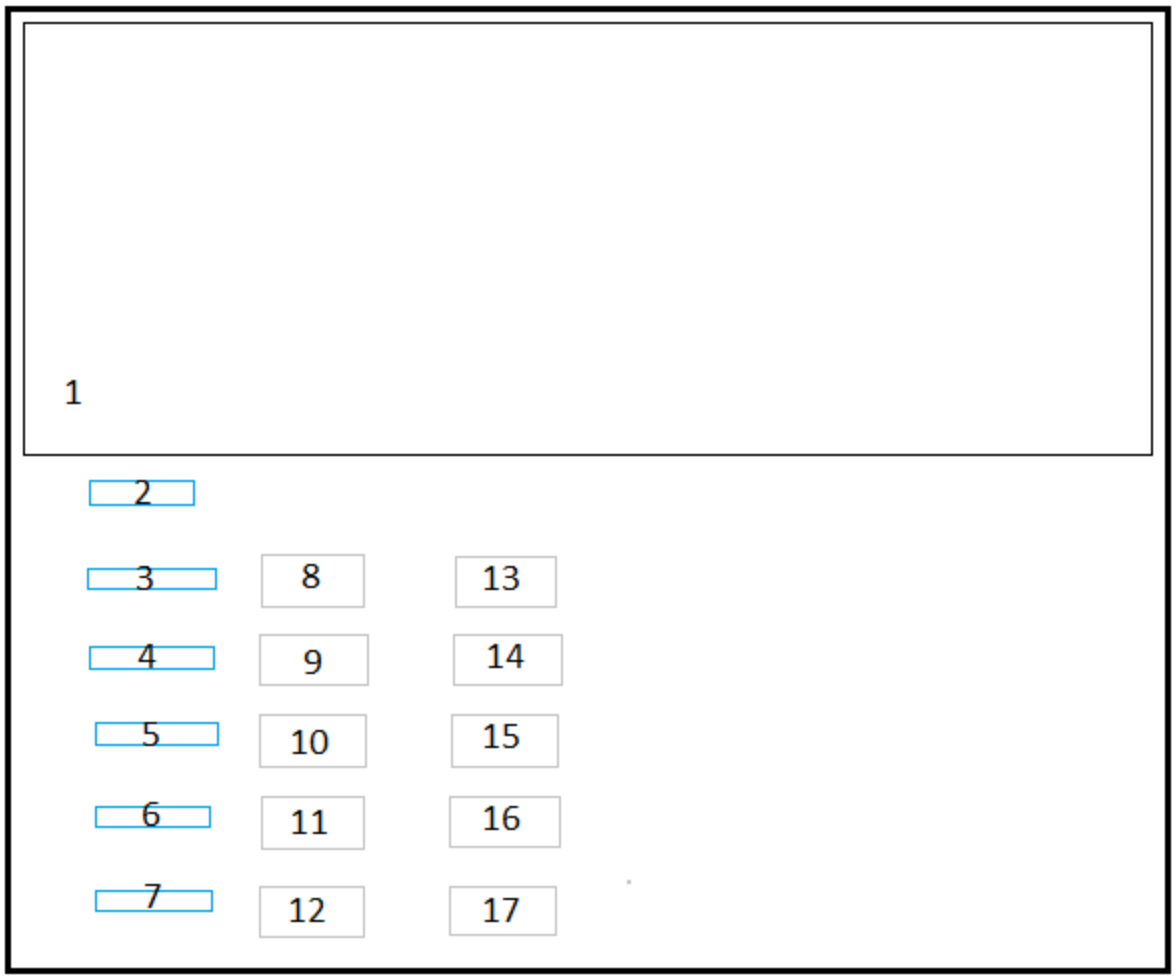
FormRank – This form will show the member's listed by their rank, it is opened when the 'Rank' button is selected on the Training Log.



Number	Object	Object Name	Text	Description	Properties
1	DataGridView	DataGridViewResults	N/A	This will display the data that is stored in the DataSet within tables	Dock - Fill
2	DataSet	DataSetResults	DataSetResults	This stores data on the training	Default

				sessions, within tables.	

FormDisplayTraining – This form displays the data from member's training sessions. This form is accessed by selecting the 'Last Days' button on the Training Log.



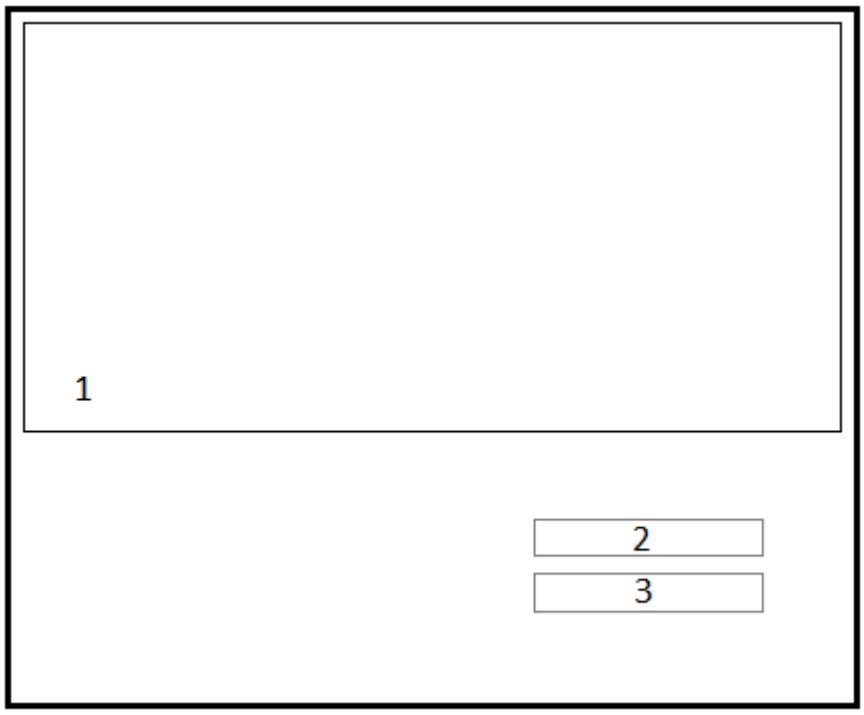
Number	Object	Object Name	Text	Description	Properties
1	DataGridView	DataGridViewMySessions	DataGridViewMySessions	This will display the data stored in DataSetClub	Default
2	Label	LabelFirstName	Label6	This will show the name of the selected	Default

				member from the training log form	
3	Label	Label1	Running	This describes what the data displayed in the following TextBoxRunningK	Default
4	Label	Label2	Cycling	This describes what the data displayed in the following TextBoxCylingK	Default
5	Label	Label3	Swimming	This describes what the data displayed in the following TextBoxSwimmingK	Default
6	Label	Label4	Total Calories	This describes what the data displayed in the following TextBoxTotal Calories	Default
7	Label	Label5	Points	This describes what the data displayed in the following TextBoxPoints	Default
8	TextBox	TextBoxRunningK	N/A	This displays the calories burned for running over a selected period of time	Default
9	TextBox	TextBoxCyclingK	N/A	This displays the calories burned for cycling over a selected period of time	Default

10	TextBox	TextBoxSwimmingK	N/A	This displays the calories burned for swimming over a selected period of time	Default
11	TextBox	TextBoxCalories	N/A	This displays the total amount of calories burned over a selected period of time	Default
12	TextBox	TextBoxPoints	N/A	This displays the number of points a member has earned over a given period of time	Default
13	TextBox	TextBoxAvRunning	N/A	This displays the average number of calories burned for running	Default
14	TextBox	TextBoxAvCycling	N/A	This displays the average number of calories burned for cycling	Default
15	TextBox	TextBoxAvSwimming	N/A	This displays the average number of calories burned for swimming	Default
16	TextBox	TextBoxAvTotalCalories	N/A	This shows the total average amount of calories burned in a given period of time	Default
17	TextBox	TextBoxAveragePoints	N/A	This displays the total average	Default

				number of points a member has earned over a given period of time	

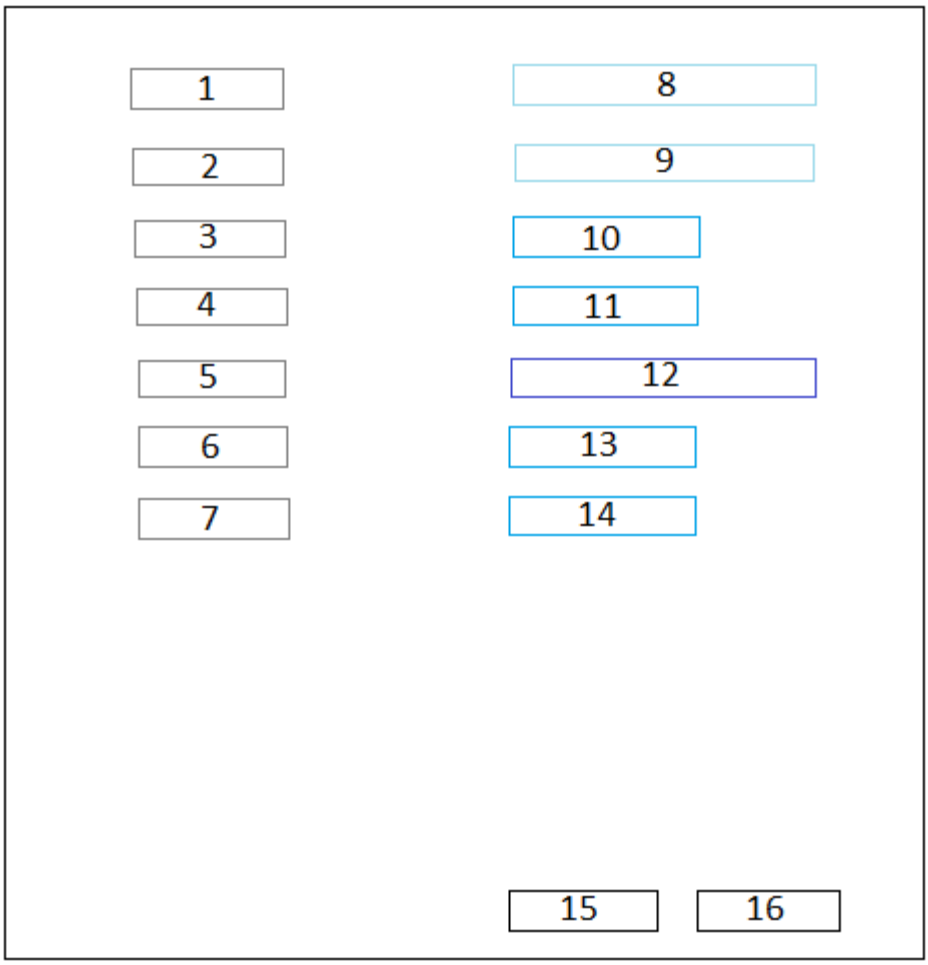
Form_Summary – This form displays the training session results for selected members. This form is accessed by selecting the 'Summary' button on the Training Log form.



Number	Object	Object Name	Text	Description	Properties
1	DataGridView	DataGridViewSummary	N/A	This will display the data which is stored in DataSetSummary	Default
2	Label	LabelTotalCalories	Total Calories	This describes what kind of	Default

				data the user is seeing displayed in the DataGridView (Total Calories)	
3	Label	LabelTotalPoints	Total Points	This describes what kind of data the user is seeing displayed in the DataGridView (TotalPoints)	Default
4	DataSet	DataSetSummary	DataSetSummary	This DataSet stores data on the calories burned and the points earned throughout training sessions.	Default

Form Training Session – This form is used by members to add a training session record to the database. The form is accessed by selecting the ‘Record a Training Session’ button on the Training Log form.



Number	Object	Object Name	Text	Description	Properties
1	Label	LabelMember	Member	Describes what the user should select in the following ComboBoxMembers	Default
2	Label	LabelActivity	Activity	Describes what the user should select in the ComboBoxActivity	Default
3	Label	LabelDuration	Duration	Describes what the user should input into the following TextBoxDuration	Default
4	Label	LabelDistance	Distance	Describes what the user should input into TextBoxDistance	Default
5	Label	LabelDate	Date	Describes what the user is supposed to	Default

				select within the DateTimePickerSession Date	
6	Label	LabelCalories	Calories	Tells the user what value is being displayed in the following TextBoxCalories	Default
7	Label	Label1	Points	Tells the user what the value being displayed is, in the following TextBoxPoints	Default
8	ComboBox	ComboBoxMembers	N/A	Here members can select members that are in the database. E.g. to record your session members will select themselves.	Default
9	ComboBox	ComboBoxActivity	N/A	This allows members to select all the sports/activities.	Default
10	TextBox	TextBoxDuration	N/A	This is where members will input their session duration time.	Default
11	TextBox	TextBoxDistance	N/A	This is where members will input their distance travelled in the training session.	Default
12	DateTimePicker	DateTimePickerSessionDate	Current date	This is for members to select the date of which their session was performed on.	Default
13	TextBox	TextBoxCalories	N/A	This TextBox will show the user how many calories they have burned in that session.	Read-Only
14	TextBox	TextBoxPoints	N/A	This TextBox will show the user how many points they have earned in their training session.	Read-Only
15	Button	ButtonOK	OK	This button when selected will save all data input, and then add this training session to the database. The form will	Default

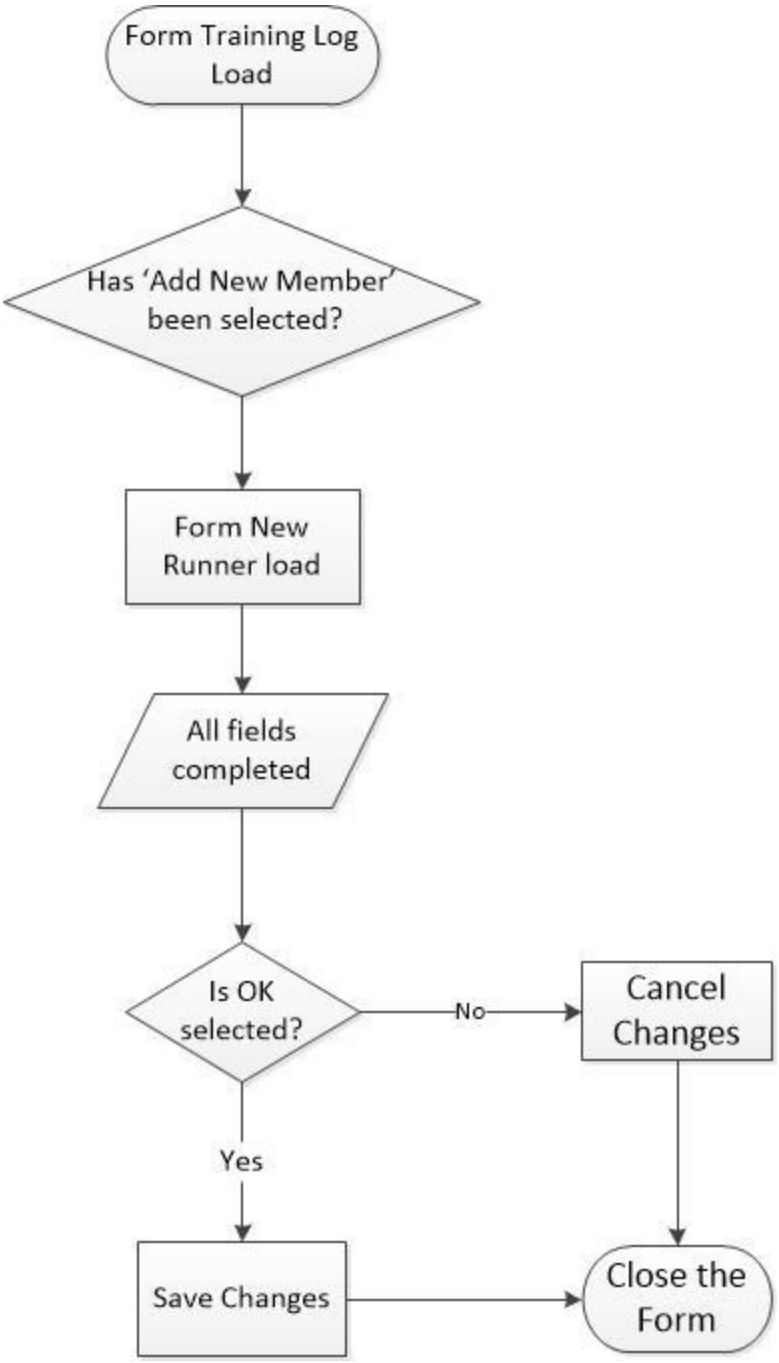
				then close after OK is selected.	
16	Button	ButtonCancel	Cancel	This button will reject all changes made and close the form.	Default

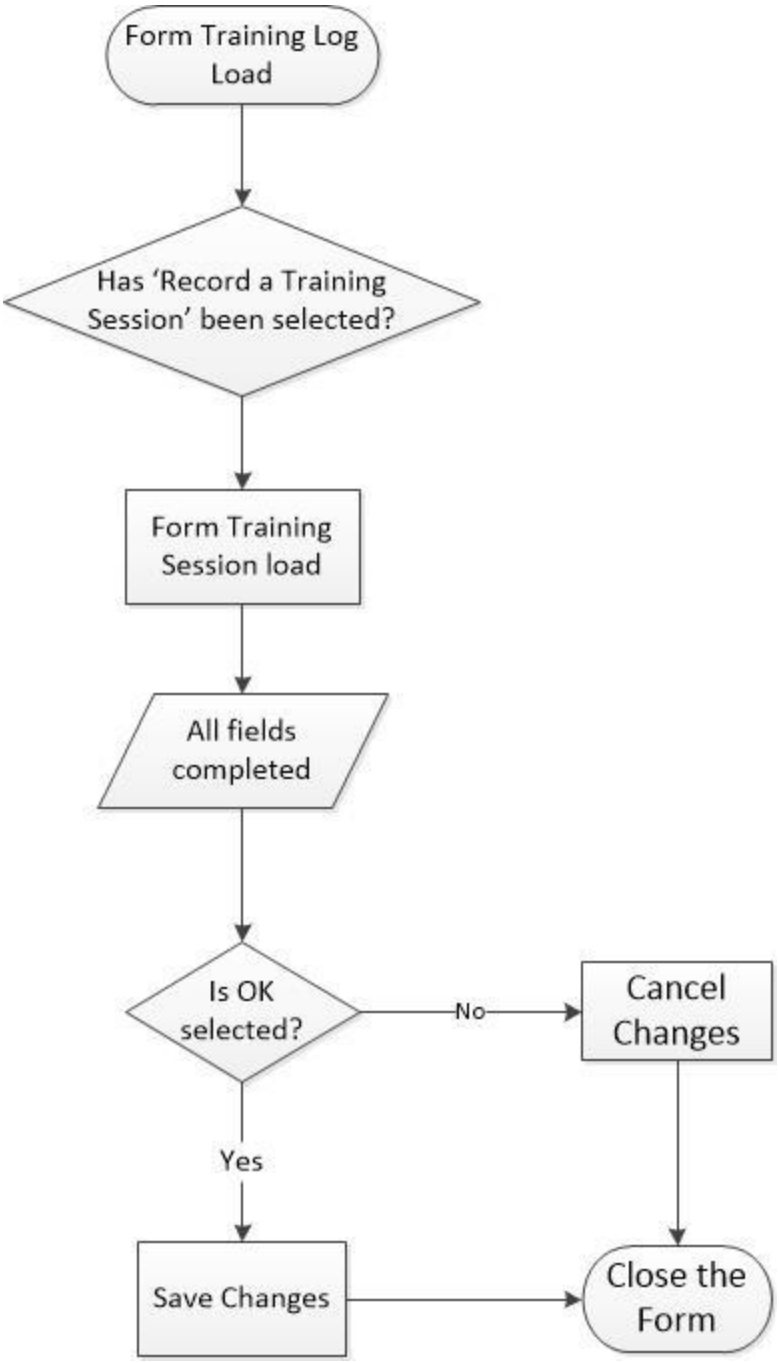
Form_Members - This form displays data within tables on all the members and all their input personal data. It is accessed by selecting the 'View Member's Details' button on the Training Log form.

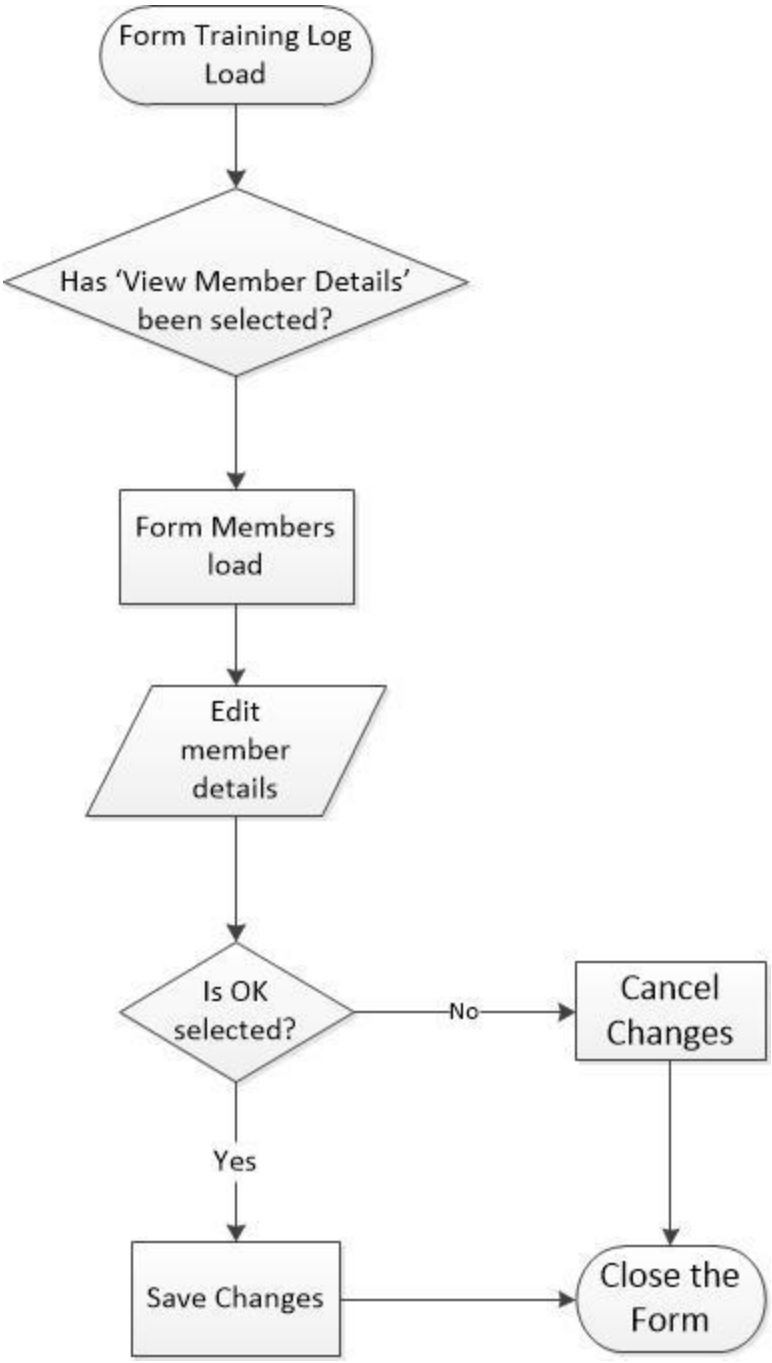
Number	Object	Object Name	Text	Description	Properties
1	DataGridView	DataGridViewMembers	N/A	This will display data which is stored in 'TableMembers' within DataSetClub	Default
2	Button	ButtonOK	OK	Any changes made to the member details will be saved once OK is selected, the	Default

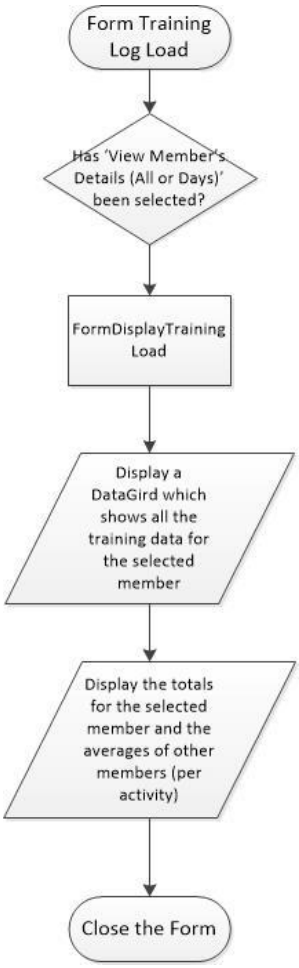
				form will then close	
3	Button	ButtonCancel	Cancel	This button rejects changes and closes the form	Default

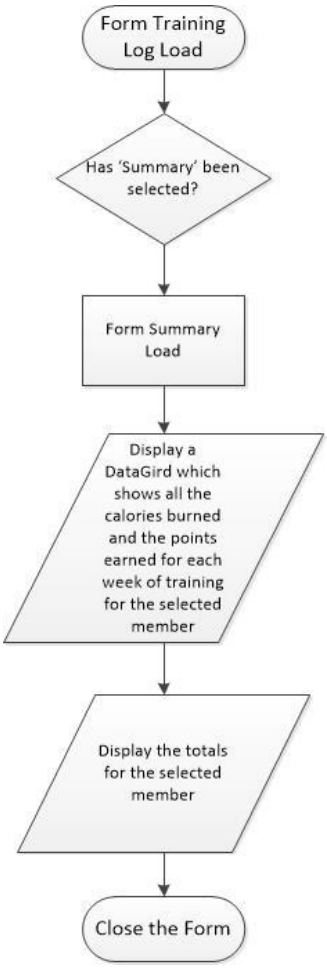
System Process Flow Diagrams

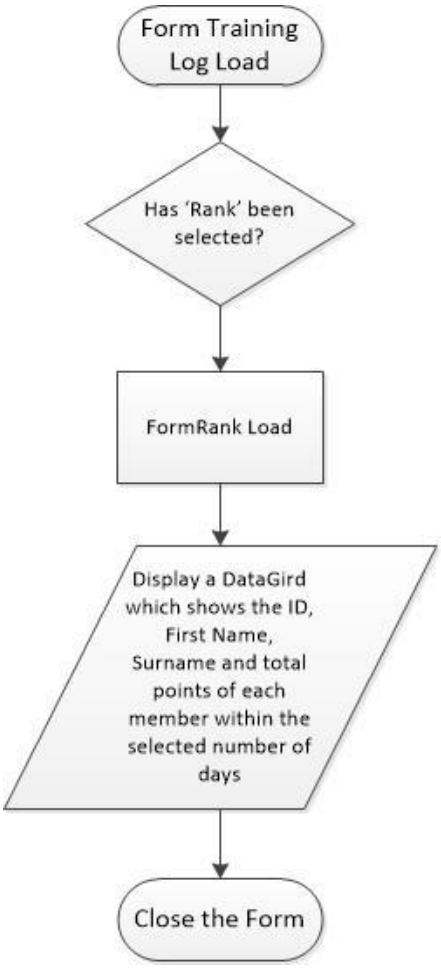

Form Training Log Load
Has 'Add New Member' been selected?
Form New Runner load
All fields completed
Is OK selected?
No
Cancel Changes
Yes
Save Changes
Close the Form


Form Training Log Load
Has 'Record a Training Session' been selected?
Form Training Session load
All fields completed
Is OK selected?
No
Cancel Changes
Yes
Save Changes
Close the Form


Form Training Log Load
Has 'View Member Details' been selected?
Form Members load
Edit member details
Is OK selected?
No
Cancel Changes
Yes
Save Changes
Close the Form


Form Training Log Load
Has 'View Member's Details (All or Days)' been selected?
FormDisplayTraining Load
Display a DataGird which shows all the training data for the selected member
Display the totals for the selected member and the averages of other members (per activity)
Close the Form


Form Training Log Load
Has 'Summary' been selected?
Form Summary Load
Display a DataGird which shows all the calories burned and the points earned for each week of training for the selected member
Display the totals for the selected member
Close the Form



Pseudo Code

Training Log

IF 'Add New Member' button is selected *THEN*

Form 'NewRunner' is loaded

ENDIF

IF 'View Member's details' button is selected *THEN*

Form Members is loaded

ENDIF

IF 'Record Training Session' button is selected *THEN*

Form Training Session is loaded

ENDIF

Display the runners' names in a combobox

IF 'All' button is selected *THEN*

FormDisplayTraining is loaded for the selected runner

ENDIF

Display numeric up/down for number of days

IF 'Last Days' button is selected *THEN*

FormDisplayTraining is loaded with for the selected runner and the selected number of days

ENDIF

IF 'Summary' button is selected *THEN*

Form_Summary is loaded

Endif

Display numeric up/down for number of days
if 'Rank' button is selected then

FormRank is loaded for the selected number of days

ENDIF

NewRunner

Read First Name

Read Surname

Read Date of Birth

Read Address

Read City

Read Postcode

Read Contact

Read Gender

Read Email Address

IF button 'Add Member' is selected *THEN*
save data to the table 'TableMembers'

ELSE

discard all changes

ENDIF

End

FormMembers

Display members' details

Read members' details

IF button 'OK' is selected *THEN*

save data to the table 'TableMembers'

ELSE

discard all changes

ENDIF

End

FormTrainingSession

Show member names in combobox

Show activity type in combobox

Read duration in minutes

Read distance in metres

Read date of the training session

Calculate calories burned

Calculate points earned

IF button 'Save' is selected THEN
save the data in the table TableTrainingSession

ELSE

discard all changes

ENDIF

End

FormDisplayTraining

Display members in DataGrid

IF member is selected

display calories burned in running

display calories burned in cycling

display calories burned in swimming

calculate total calories burned

display total calories burned

display total points earned

ENDIF

End

Form Summary

Display data from the 'DataSetSummary' in the DataGridView

Display total calories burned

Display total points earned

ENDIF

End

Form Rank

Display data from the 'DataSetResults' within the DataSetView

ENDIF

End

Evaluation Criteria

Usability

This program must be easy to use and require little IT experience in order to navigate through the system efficiently. The usability of this application can be tested by asking people with little IT experience to use the program. When they have tried out the program I will be able to gain feedback by asking them to complete a short questionnaire which will ask relevant questions in order to gain a user's evaluation of this computer program. I will use the following questions in the user questionnaire:

- Did you find the program easy or difficult to use?

  Very easy ☐

  Easy ☐

  Hard ☐

  Very hard ☐

- Did you need any help with any of the tasks?

  None ☐

  For some things ☐

  For most things ☐

  ☐

For all tasks

If the majority of users found it easy to use with little or no difficulty then this will be the criteria for a successful program that requires no additional modification, allowing us to evaluate the usability.

Stability

The system must be stable so that no data is lost as a result of errors. This can be fixed by testing the system thoroughly by applying logical and illogical sequences, such as entering no values in text boxes, etc. IN order to evaluate the stability I will ask users to test the system and then I will give them a questionnaire to fill out with a simple question that should efficiently evaluate the stability. I will ask the following question to determine the stability of my system:

- Whilst using the program did you find any problems? (Errors, crashing, etc)

Yes ☐

No ☐

Usefulness

Perhaps most importantly the computer program has to be useful and meet the given requirements. In order to better understand the usefulness of our program I will give users the following simple question to help us determine the usefulness

- How useful do you find the computer program?

Very useful ☐

Quite useful ☐

Not very useful ☐

Not useful at all ☐

Success will be determined by a ‘Quite useful’ – ‘Very useful’ average. Given this criteria the program will be ready for actual use by Parkwood Vale Harriers.

Performance

The computer program must excel at what it is designed for. This means it must be fast and must be reliable enough for members to use this consistently on a regular basis. Users should be able to transition between forms and perform calculations with very little to no waiting time whatsoever.

In order to test the quality of performance I will give users this final questionnaire in order to evaluate the program’s performance

- How much loading time did you experience whilst using the program?

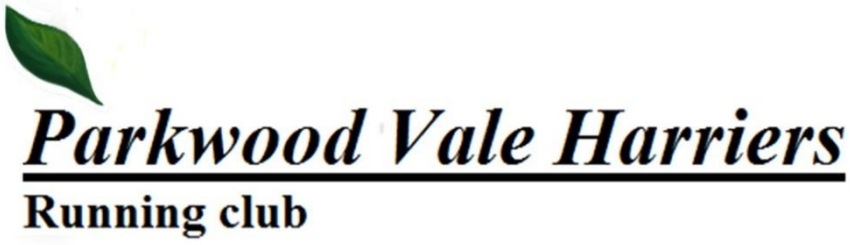
  None ☐

  A little ☐

  Some ☐

  A lot ☐

Success in this field will be considered successful if users have an average answer of ‘None’ in this questionnaire.
Then I will consider this program to be fully functional and ready for the Parkwood Vale Harriers to begin implementation of the computer program.

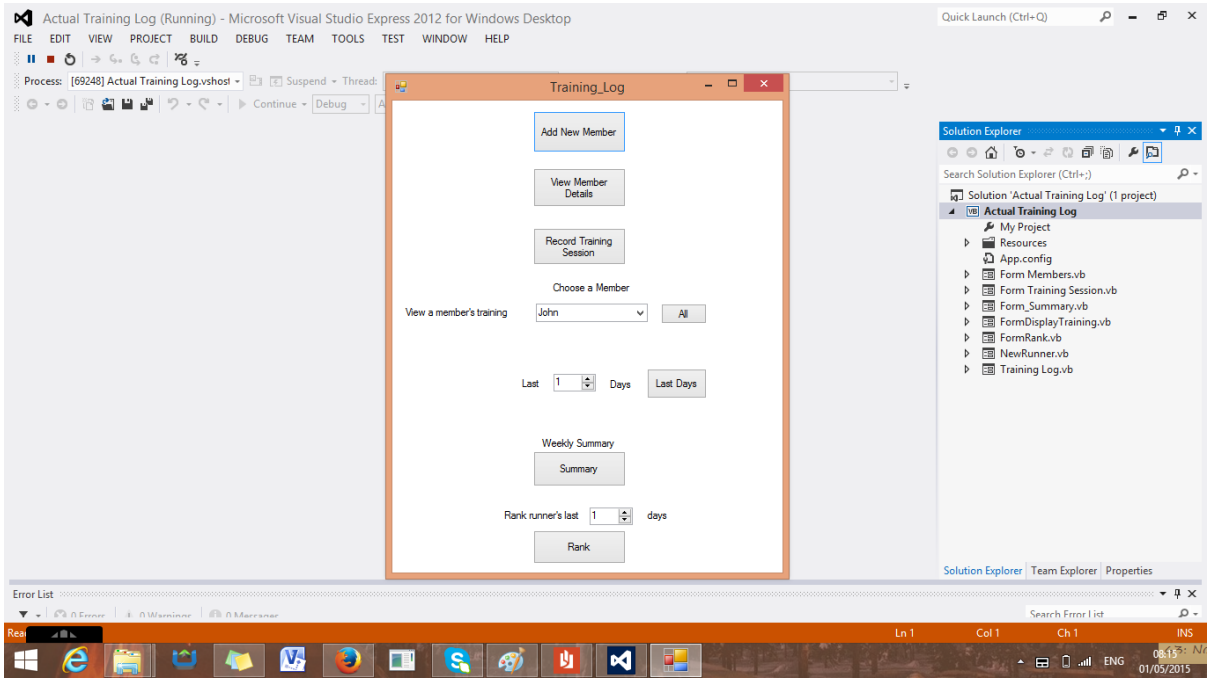


Software Development

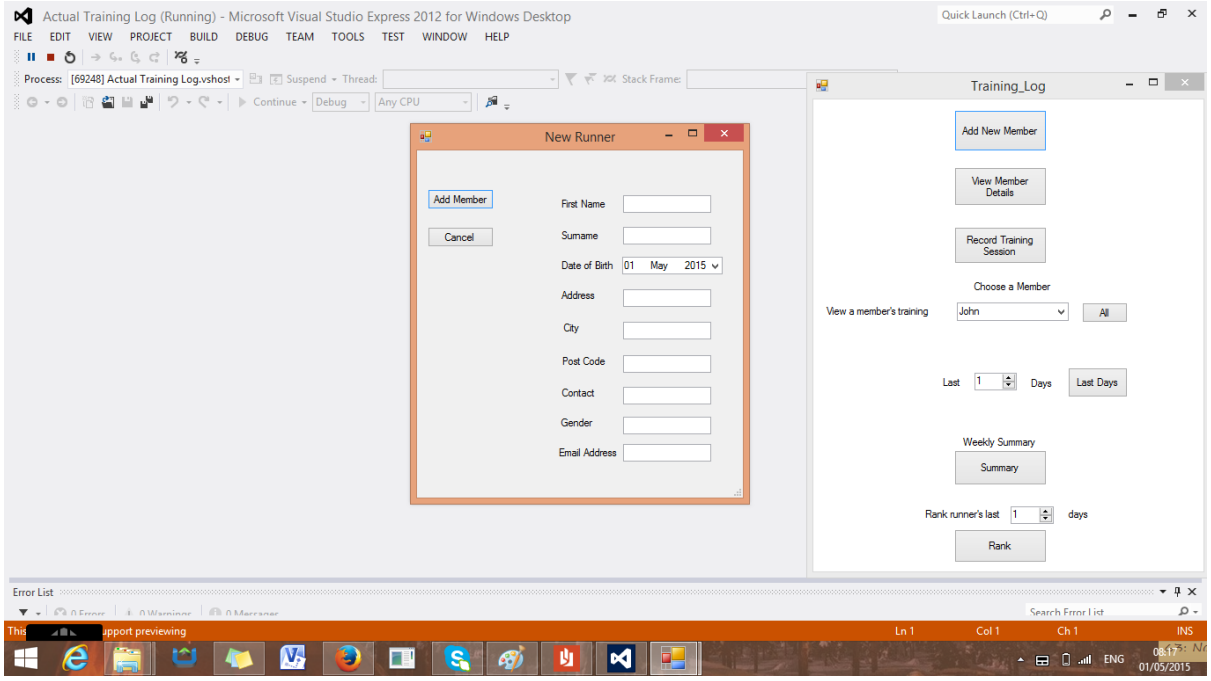
Program Documentation

User Interface

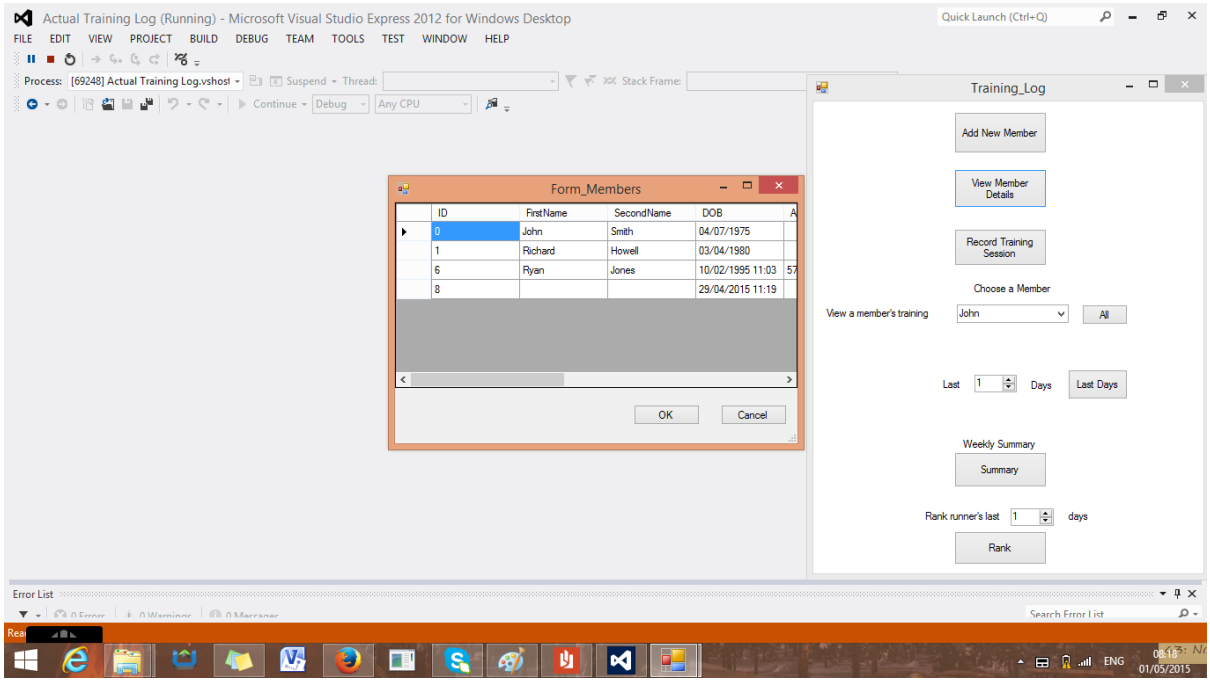
TrainingLog



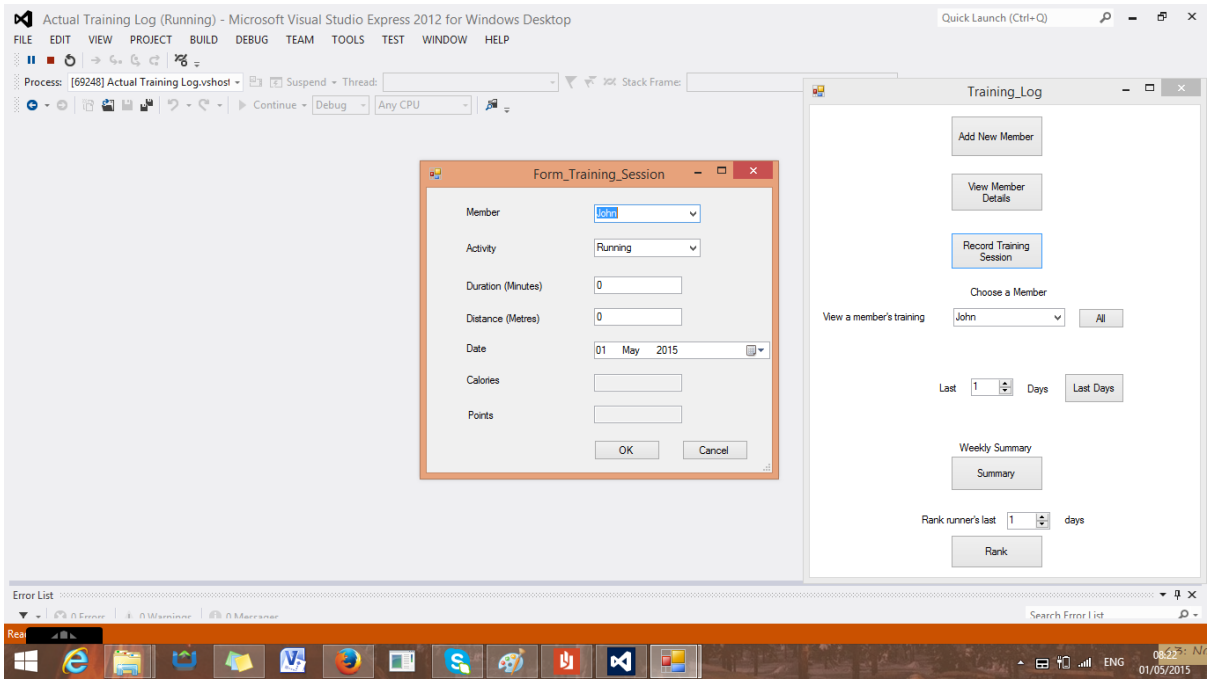
NewRunner



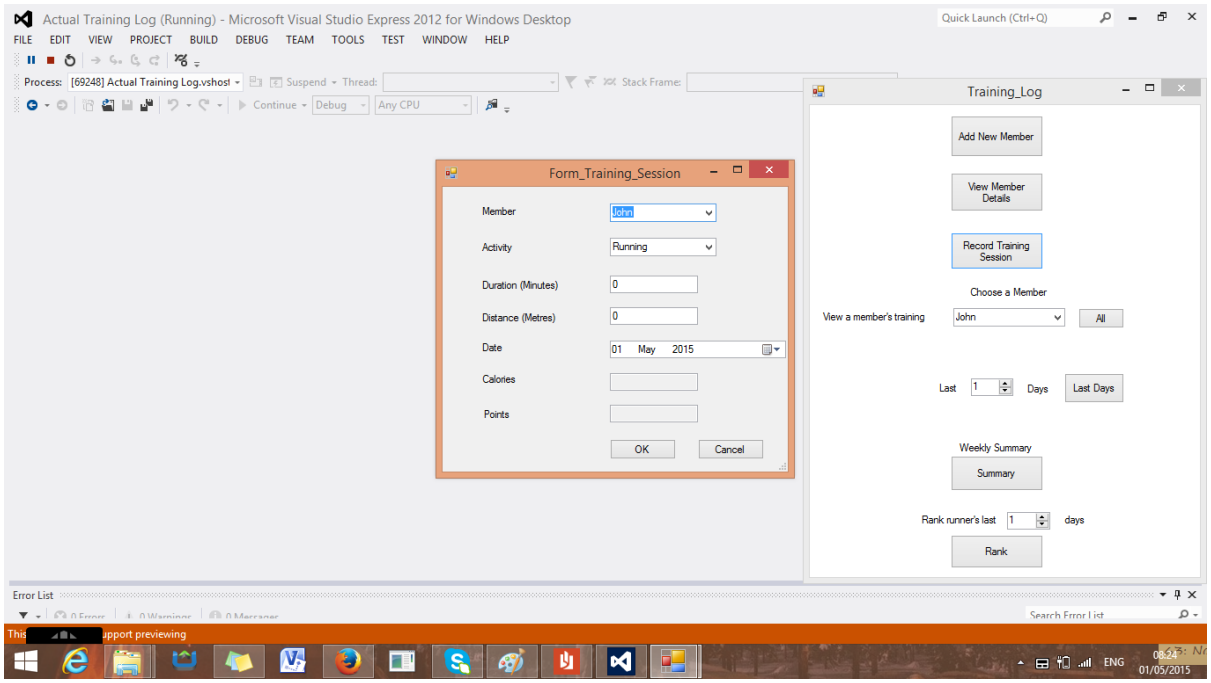
Form_Members



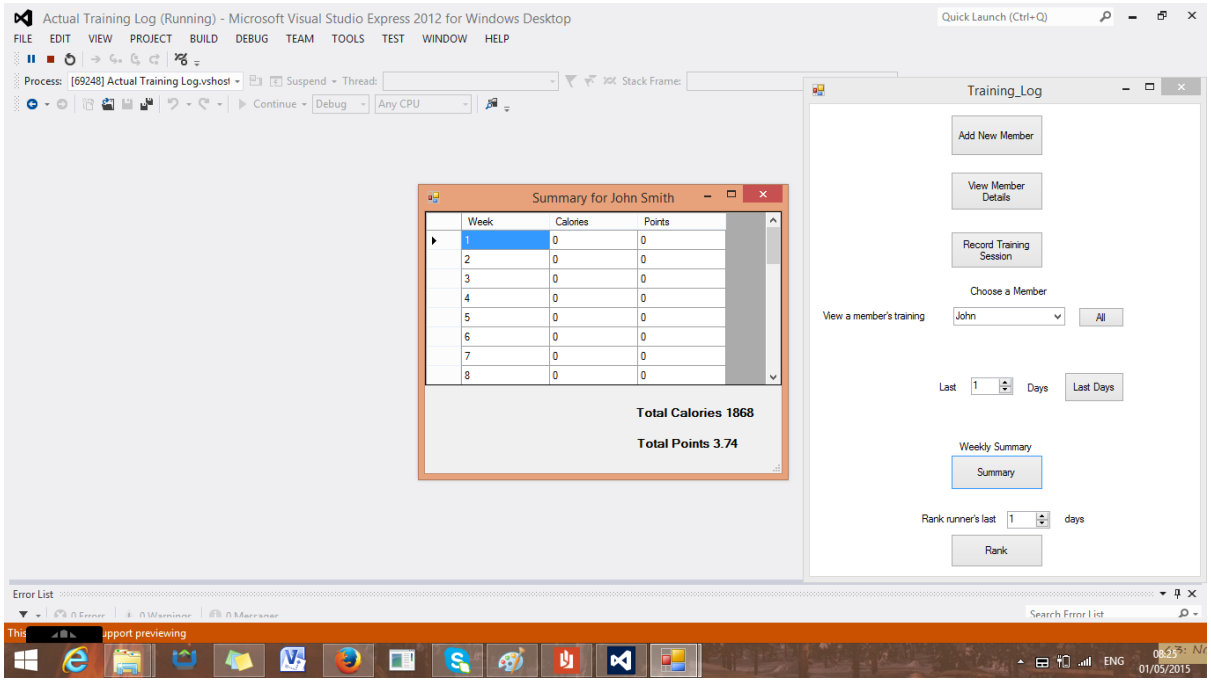
Form_Training_Session



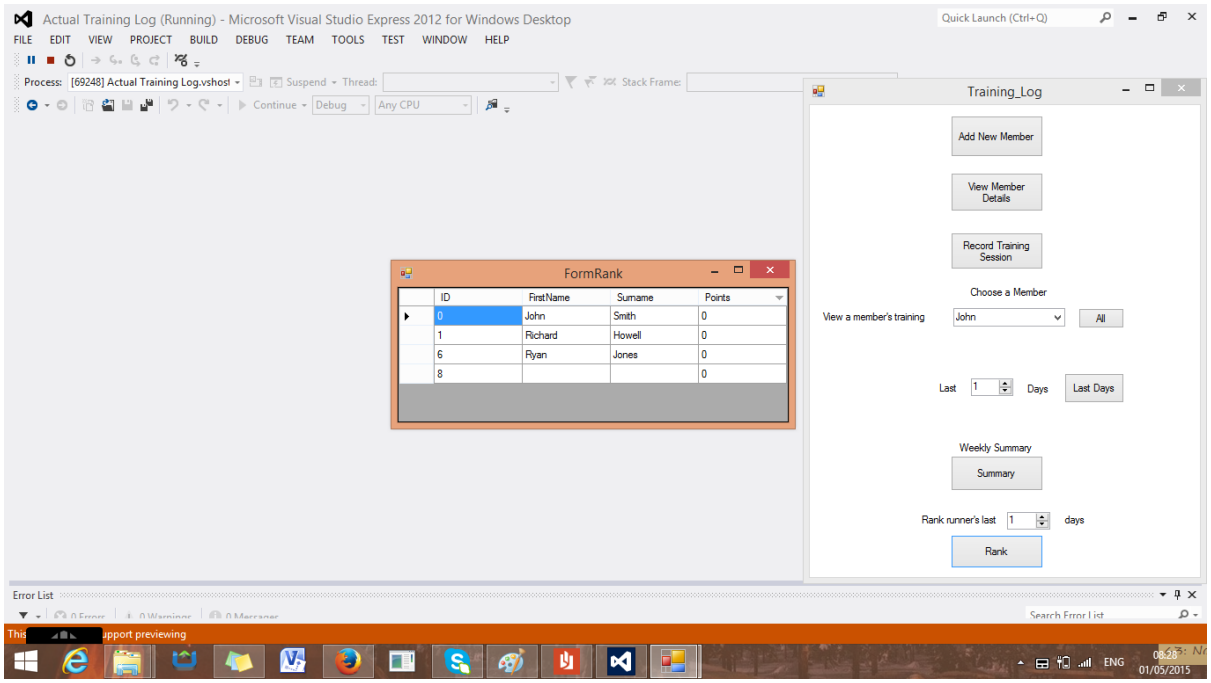
FormDisplayTraining



Form_Summary



FormRank



Programme Code

Training Log

```
Public Class Training_Log
    'Format requirements for DATE: enclose a date literak within number signs (# #).
    'specify the date value in the formant M/d/yyyy, for example #5/31/1993#.
    'This requirement is independent of your locale and your computer's data and time format settings.
    Friend startdate As Date = #10/1/2014#

    Friend savetable As Boolean = False

    Private Sub ButtonAddNewMember_Click(sender As Object, e As EventArgs) Handles ButtonAddNewMember.Click
        Dim f As New NewRunner
        f.ds = DataSetClub
        If (f.ShowDialog() = Windows.Forms.DialogResult.OK) Then
            savetable = True
            DataSetClub.AcceptChanges()
        End If

    End Sub

    Private Sub Training_Log_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
        Try

            DataSetClub.ReadXml("dataset.xml")

        Catch ex As Exception

            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "Running")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "Cycling")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "Freestyle slow")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "FreeStyle fast")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "Backstroke")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "Breaststroke")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "Butterfly")
            savetable = True

        End Try

        DataSetClub.AcceptChanges()

        ComboBoxMembers.DataSource = DataSetClub.Tables("TableMembers")
        ComboBoxMembers.DisplayMember = "FirstName"
        ComboBoxMembers.ValueMember = "ID"

        Dim d As Date = Date.Today
        Dim t As TimeSpan = d.Subtract(startdate)

        NumericUpDownRank.Minimum = 1
        NumericUpDownRank.Maximum = t.Days()

        NumericUpDownDays.Minimum = 1
        NumericUpDownDays.Maximum = t.Days()

    End Sub

    Private Sub Training_Log_FormClosed(sender As Object, e As FormClosedEventArgs)
Handles MyBase.FormClosed
        If savetable = True Then
            DataSetClub.WriteXml("dataset.xml")
        End If

    End Sub

    Private Sub ButtonViewMemberDetails_Click(sender As Object, e As EventArgs) Handles
ButtonViewMemberDetails.Click

        Dim f As New FormMembers
        f.ds = DataSetClub ' Comment from Negem: You forgot to pass the dataset to the new
```

```
form
        If (f.ShowDialog() = Windows.Forms.DialogResult.OK) Then
            savetable = True
            DataSetClub.AcceptChanges()
        End If

    End Sub

    Private Sub ButtonRecordTrainingSession_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ButtonRecordTrainingSession.Click

        Dim f As New Form_Training_Session

        f.ds = DataSetClub

        If (f.ShowDialog() = DialogResult.OK) Then
            DataSetClub.Tables("TableTrainingSession").Rows.Add(Nothing,
            f.ComboBoxMembers.SelectedValue, f.ComboBoxActivity.SelectedValue,
            f.TextBoxDuration.Text, f.TextBoxDistance.Text,
            f.DateTimePickerSessionDate.Value.Date, f.TextBoxCalories.Text,
f.TextBoxPoints.Text)
            savetable = True

            DataSetClub.AcceptChanges()

        End If

    End Sub

    Private Sub ButtonByWeek_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonByWeek.Click
        Dim f As New Form_Summary
        Dim drv As DataRowView = ComboBoxMembers.SelectedItem
        f.Text = "Summary for " + drv("FirstName") + " " + drv("SecondName")
        f.ds = DataSetClub
        f.memberid = ComboBoxMembers.SelectedValue
        f.startd = startDate
        f.ShowDialog()

    End Sub


    Private Sub ButtonRank_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonRank.Click

        Dim f As New FormRank
```

```
        f.days = NumericUpDownRank.Value
        f.ds = DataSetClub
        f.ShowDialog()

    End Sub


    Private Sub ButtonDays_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonDays.Click

        Dim f As New FormDisplayTraining
        Dim drv As DataRowView = ComboBoxMembers.SelectedItem
        f.Text = "Training record of " & drv("FirstName") & " " & drv("SecondName")
        f.days = NumericUpDownDays.Value
        f.ds = DataSetClub
        f.memberID = ComboBoxMembers.SelectedValue
        f.LabelFirstName.Text = drv("FirstName")
        f.ShowDialog()

    End Sub

    Private Sub ButtonView_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonView.Click
        Dim f As New FormDisplayTraining
        Dim drv As DataRowView = ComboBoxMembers.SelectedItem
        f.Text = "Training record of " & drv("FirstName") & " " &
        drv("SecondName")
        f.LabelFirstName.Text = drv("FirstName")
        f.ds = DataSetClub
        f.memberID = ComboBoxMembers.SelectedValue
        f.ShowDialog()
    End Sub


End Class
```

NewRunner

```
Public Class NewRunner

    Friend ds As DataSet

    Private Sub ButtonAddMember_Click(sender As Object, e As EventArgs) Handles
ButtonAddMember.Click

        ds.Tables("TableMembers").Rows.Add(Nothing, TextBoxFirstName.Text,
```

```
TextBoxSurname.Text, DateTimePickerDOB.Value, TextBoxAddress.Text,
TextBoxCity.Text, TextBoxPostCode.Text, TextBoxContact.Text, TextBoxGender.Text,
TextBoxEmailAddress.Text)

        Me.DialogResult = DialogResult.OK

    End Sub

    Private Sub ButtonCancel_Click(sender As Object, e As EventArgs) Handles
ButtonCancel.Click

        Me.DialogResult = DialogResult.Cancel

    End Sub

    Private Sub TextBoxSurname_TextChanged(sender As Object, e As EventArgs) Handles
TextBoxSurname.TextChanged

    End Sub
End Class
```

Form_Training_Session

```
Public Class Form_Training_Session
    Friend ds As DataSet
    Private loaded As Boolean = False



    Private Function calculatecalories(ByVal activityid As Integer, ByVal minutes As Integer,
ByVal metres As Integer) As Integer

        Dim result As Integer = 0
        Dim speedmph As Decimal

        Dim cph As Integer = 0

        If (activityid = 0) Then 'running

            speedmph = (TextBoxDistance.Text / TextBoxDuration.Text) * 0.0372822715    'mph

            If (speedmph <= 5) Then

                cph = 472
```

```
        ElseIf (speedmph < 6) Then
            cph = 590
        ElseIf (speedmph < 7) Then
            cph = 679
        ElseIf (speedmph < 8) Then
            cph = 797
        ElseIf (speedmph < 9) Then
            cph = 885
        Else
            cph = 944
        End If
        result = cph * (TextBoxDuration.Text / 60)
        Return result
    End If
    If (activityid = 1) Then 'cycling
        speedmph = (TextBoxDistance.Text / TextBoxDuration.Text) * 0.0372822715    'mph
        If (speedmph < 10) Then
            cph = 236
        ElseIf (speedmph < 12) Then
            cph = 354
        ElseIf (speedmph < 14) Then
            cph = 472
        ElseIf (speedmph < 16) Then
```

```
            cph = 590
        ElseIf (speedmph < 20) Then
            cph = 708
        Else
            cph = 944
        End If
        result = cph * (TextBoxDuration.Text / 60)
        Return result
    End If
    Dim hours As Decimal = TextBoxDuration.Text / 60
    If (activityid = 2) Then 'front crawl slow
        Return hours * 413
    End If
    If (activityid = 3) Then 'front crawl fast
        Return hours * 590
    End If
    If (activityid = 4) Then 'backstroke
        Return hours * 413
    End If
    If (activityid = 5) Then 'breaststroke
        Return hours * 590
    End If
    If (activityid = 6) Then 'butterfly
```

```
            Return hours * 649

        End If

        Return result

    End Function

    Private Function calculatepoints(ByVal activityid As Integer, ByVal calories As Integer,
ByVal sessionTime As Integer, ByVal sessiondistance As Integer) As Decimal

        Dim result As Decimal = 0

        result = result + calories / 500

        If (activityid = 0) Then

            If sessionTime >= 60 Then

                Dim mph As Decimal = (sessiondistance / sessionTime) * 0.0372822715

                If mph > 6 Then result = result + (mph - 6) / 6

            End If

        End If

        Return result

    End Function

    Private Sub ButtonOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonOK.Click

        ' Make sure that the data is valid before making the calculations

        Dim ValidData As Boolean = True

        If DateTimePickerSessionDate.Value.Date > Date.Today Or
DateTimePickerSessionDate.Value.Date < #10/1/2014# Then

            MessageBox.Show("Invalid Date")

            ValidData = False

        End If
```

```
        If TextBoxDistance.Text = "0" Then

            MessageBox.Show("Invalid Distance")

            ValidData = False

        End If

        If TextBoxDuration.Text = "0" Then

            MessageBox.Show("Invalid Duration")

            ValidData = False

        End If

        If ValidData Then

            Me.DialogResult = DialogResult.OK

        End If

    End Sub

    Private Sub ButtonCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ButtonCancel.Click

        Me.DialogResult = DialogResult.Cancel

    End Sub

    Private Sub FormTrainingSession_FormClosing(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

        'The Calories and Points textboxes are filled in on form closing, and could well be
invisible

        If Me.DialogResult = DialogResult.OK Then
            TextBoxCalories.Text =
calculatecalories(ComboBoxActivity.SelectedValue,TextBoxDuration.Text,
TextBoxDistance.Text)
            TextBoxPoints.Text = calculatepoints(ComboBoxActivity.SelectedValue,
            TextBoxCalories.Text, TextBoxDuration.Text, TextBoxDistance.Text)

        End If
```

```
    End Sub


    Private Sub Form_Training_Session_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        ComboBoxMembers.DataSource = ds.Tables("TableMembers")
        ComboBoxMembers.DisplayMember = "FirstName"
        ComboBoxMembers.ValueMember = "ID"

        ComboBoxActivity.DataSource = ds.Tables("TableSports")
        ComboBoxActivity.DisplayMember = "Description"
        ComboBoxActivity.ValueMember = "ID"

        loaded = True

    End Sub
End Class
```

FormDisplayTraining

```
Public Class FormDisplayTraining

    Friend ds As DataSet
    Friend memberID As Integer
    Friend days As Integer = 0
    Private bs As BindingSource

    Private Function calculatecalories(ByVal activityid As Integer) As Integer
        Dim result As Integer = 0

        Dim drv As DataRowView


        For Each o As Object In bs.List
            drv = o
            If (drv("sportID") = activityid) Then result = result + drv("CalorificValue")
        Next
        Return result
    End Function

    Private Function calculatepoints() As Decimal
        Dim result As Decimal = 0

        Dim drv As DataRowView
```

```
        For Each o As Object In bs.List
            drv = o
            result = result + drv("points")
        Next
        Return result
    End Function

    Private Sub FormDisplayTraining_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        bs = New BindingSource
        bs.DataSource = ds.Tables("TableTrainingSession")
        bs.Filter = "memberID = " & memberID

        Dim d As Date = Date.Today
        Dim dstr As String

        If days > 0 Then

            d = d.AddDays(-days)
            dstr = d.Month & "/" & d.Day & "/" & d.Year

            bs.Filter += " AND  SessionDate >= #" & dstr & "#"

        End If

        ' DataGridViewMySesssions.DataSource = bs Comments from Negem: the name of the
DataGrid is wrong Sesssions
        DataGridViewMySessions.DataSource = bs

        TextBoxRunningK.Text = calculatecalories(0)
        TextBoxCyclingK.Text = calculatecalories(1)

        TextBoxSwimmingK.Text = calculatecalories(2) + calculatecalories(3) +
calculatecalories(4) +
        calculatecalories(5) + calculatecalories(6)

        TextBoxCalories.Text = calculatecalories(0) + calculatecalories(1) +
calculatecalories(2) +
        calculatecalories(3) + calculatecalories(4) + calculatecalories(5) + calculatecalories(6)

        TextBoxPoints.Text = Format(calculatepoints(), "N")
        TextBoxAvRunning.Text = Format(calculateAverage(0, days), "N")
        TextBoxAvCycling.Text = Format(calculateAverage(1, days), "N")
        TextBoxAvSwimming.Text = Format(calculateAverageSwim(days), "N")
        TextBoxAvTotalCalories.Text = Format(calculateAvAllActivities(days), "N")
```

```
        TextBoxAveragePoints.Text = Format(calculateAvPoints(days), "N")

    End Sub

    Private Function calculateAverage(ByVal activityid As Integer, ByVal days As Integer) As
Decimal

        Dim d As Date = Date.Today

        d = d.AddDays(-days)

        Dim tot As Decimal = 0

        Dim l As New ArrayList

        For Each dr As DataRow In ds.Tables("TableTrainingSession").Rows

            If dr("memberID") <> memberID And dr("sportid") = activityid And (days =
            0 Or dr("SessionDate") >= d) Then

                tot = tot + dr("CalorificValue")
                If Not l.Contains(dr("memberid")) Then l.Add(dr("memberid"))

            End If

        Next

        If (l.Count <> 0) Then Return tot / l.Count

        Return 0

    End Function

    Private Function calculateAverageSwim(ByVal days As Integer) As Decimal

        Dim d As Date = Date.Today

        d = d.AddDays(-days)

        Dim tot As Decimal = 0

        Dim l As New ArrayList

        For Each dr As DataRow In ds.Tables("TableTrainingSession").Rows

            If dr("memberID") <> memberID And dr("sportid") >= 2 And (days = 0 Or
```

```
            dr("SessionDate") >= d) Then

                tot = tot + dr("CalorificValue")
                If Not l.Contains(dr("memberid")) Then l.Add(dr("memberid"))

            End If

        Next

        If (l.Count <> 0) Then Return tot / l.Count

        Return 0

    End Function

    Private Function calculateAvAllActivities(ByVal days As Integer) As Decimal

        Dim d As Date = Date.Today

        d = d.AddDays(-days)

        Dim tot As Decimal = 0

        Dim l As New ArrayList

        For Each dr As DataRow In ds.Tables("TableTrainingSession").Rows

            If dr("memberID") <> memberID And (days = 0 Or dr("SessionDate") >= d) Then

                tot = tot + dr("CalorificValue")
                If Not l.Contains(dr("memberid")) Then l.Add(dr("memberid"))

            End If

        Next

        If (l.Count <> 0) Then Return tot / l.Count

        Return 0

    End Function

    Private Function calculateAvPoints(ByVal days As Integer) As Decimal

        Dim d As Date = Date.Today
```

```
        d = d.AddDays(-days)

        Dim tot As Decimal = 0

        Dim l As New ArrayList

        For Each dr As DataRow In ds.Tables("TableTrainingSession").Rows

            If dr("memberID") <> memberID And (days = 0 Or dr("SessionDate") >= d) Then

                tot = tot + dr("Points")
                If Not l.Contains(dr("memberid")) Then l.Add(dr("memberid"))

            End If

        Next

        If (l.Count <> 0) Then Return tot / l.Count

        Return 0

    End Function

    Private Sub DataGridViewMySessions_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs) Handles DataGridViewMySessions.CellContentClick

    End Sub
End Class
```

FormSummary

```
Public Class Form_Summary


    Friend ds As DataSet
    Friend memberid As Integer
    Friend startd As Date
    Private bs As BindingSource

    Private Function calccals() As Integer
        Dim result As Integer = 0
        Dim drv As DataRowView
        For Each o As Object In bs.List
            drv = o
```

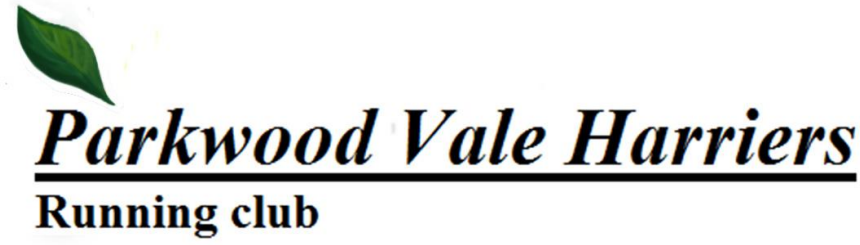
```
            result = result + drv("CalorificValue")
        Next
        Return result
    End Function

    Private Function calcpoints() As Decimal
        Dim result As Decimal = 0
        Dim drv As DataRowView
        For Each o As Object In bs.List
            drv = o
            result = result + drv("points")
        Next
        Return result
    End Function

    Private Sub FormSummary_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        bs = New BindingSource
        bs.DataSource = ds.Tables("TableTrainingSession")
        Dim done As Boolean = False
        Dim d As Date = startd
        Dim weeknumber As Integer = 0
        Dim totalCalories As Integer = 0
        Dim totalPoints As Decimal = 0.0
        Do While Not done
            weeknumber = weeknumber + 1
            bs.Filter = "memberID = " & memberid
            Dim d1 As Date = d
            Dim dstr1 As String = d1.Month & "/" & d1.Day & "/" & d1.Year
            d = d.AddDays(7)
            Dim dst As String = d.Month & "/" & d.Day & "/" & d.Year
            If d >= Date.Today Then done = True
            Try
                bs.Filter += " AND SessionDate >= #" & dstr1 & "# AND SessionDate <= #" & dst
& "#"
                DataSetSummary.Tables(0).Rows.Add(weeknumber, calccals, calcpoints)
                totalCalories = totalCalories + calccals()
                totalPoints = totalPoints + calcpoints()
            Catch ex As Exception
                MessageBox.Show(ex.Message & " " & bs.Filter)
            End Try
        Loop
        DataGridViewSummary.DataSource = DataSetSummary.Tables(0)
        LabelTotalCalories.Text += " " & totalCalories
        LabelTotalPoints.Text += " " & Format(totalPoints, "N")
```

```
    End Sub


End Class
```

FormRank

```
Public Class FormRank

    Friend ds As DataSet
    Friend days As Integer = 0

    Private Function points(ByVal days As Integer, ByVal memberID As Integer) As Double

        Dim bs As New BindingSource
        bs.DataSource = ds.Tables("TableTrainingSession")
        bs.Filter = "memberID = " & memberID

        Dim d As Date = Date.Today
        Dim dstr As String

        d = d.AddDays(-days)
        dstr = d.Month & "/" & d.Day & "/" & d.Year

        bs.Filter += " AND  SessionDate >= #" & dstr & "#"
        Dim result As Double = 0
        Dim i As Integer
        For i = 0 To 6
            result = result + calculatepoints(i, bs)
        Next
        Return result

    End Function

    Private Function calculatepoints(ByVal activityid As Integer, ByVal bs As BindingSource)
As Double

        Dim result As Double = 0
        Dim drv As DataRowView

        For Each o As Object In bs.List
            drv = o
            If (drv("sportID") = activityid) Then result = result +
            drv("CalorificValue") / 500
            If (drv("sportID") = 0) Then
```

```
            If drv("SessionTime") >= 60 Then
                Dim mph As Double = drv("SessionDistance") / drv("SessionTime") *
0.0372822715
                If mph > 6 Then result = result + (mph - 6) / 6
            End If
        End If
    Next
    Return result

End Function

Private Sub FormRank_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    Dim bs As New BindingSource
    bs.DataSource = ds.Tables("TableTrainingSession")
    For Each dr As DataRow In ds.Tables("TableMembers").Rows
        If DataSetResults.Tables(0).Rows Is Nothing Then MessageBox.Show("null")
        Try
            DataSetResults.Tables(0).Rows.Add(dr("ID"), dr("FirstName"),
            dr("SecondName"), points(days, dr("ID")))
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    Next
    DataSetResults.AcceptChanges()
    DataGridViewResults.DataSource = DataSetResults.Tables(0)
    DataGridViewResults.Sort(DataGridViewResults.Columns("Points"),
    System.ComponentModel.ListSortDirection.Descending)

End Sub

Private Sub DataGridViewResults_CellContentClick(sender As Object, e As
DataGridViewCellEventArgs) Handles DataGridViewResults.CellContentClick

End Sub
End Class
```



Software Development

Program Code

Training Log

```
Public Class Training_Log
    'Format requirements for DATE: enclose a date literak within number signs (# #).
    'specify the date value in the formant M/d/yyyy, for example #5/31/1993#.
    'This requirement is independent of your locale and your computer's data and time format settings.
    Friend startdate As Date = #10/1/2014#

    Friend savetable As Boolean = False

    Private Sub ButtonAddNewMember_Click(sender As Object, e As EventArgs) Handles ButtonAddNewMember.Click
        Dim f As New NewRunner
        f.ds = DataSetClub
        If (f.ShowDialog() = Windows.Forms.DialogResult.OK) Then
            savetable = True
            DataSetClub.AcceptChanges()
        End If

    End Sub

    Private Sub Training_Log_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        Try

            DataSetClub.ReadXml("dataset.xml")

        Catch ex As Exception
```

```
            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "Running")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "Cycling")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "Freestyle
slow")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "FreeStyle
fast")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing,
"Backstroke")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing,
"Breaststroke")
            DataSetClub.Tables("TableSports").Rows.Add(Nothing, "Butterfly")
            savetable = True

        End Try

        DataSetClub.AcceptChanges()

        ComboBoxMembers.DataSource =
DataSetClub.Tables("TableMembers")
        ComboBoxMembers.DisplayMember = "FirstName"
        ComboBoxMembers.ValueMember = "ID"

        Dim d As Date = Date.Today
        Dim t As TimeSpan = d.Subtract(startdate)

        NumericUpDownRank.Minimum = 1
        NumericUpDownRank.Maximum = t.Days()

        NumericUpDownDays.Minimum = 1
        NumericUpDownDays.Maximum = t.Days()

    End Sub

    Private Sub Training_Log_FormClosed(sender As Object, e As
```

```
FormClosedEventArgs) Handles MyBase.FormClosed
        If savetable = True Then
            DataSetClub.WriteXml("dataset.xml")
        End If

    End Sub

    Private Sub ButtonViewMemberDetails_Click(sender As Object, e As
EventArgs) Handles ButtonViewMemberDetails.Click

        Dim f As New FormMembers
        f.ds = DataSetClub ' Comment from Negem: You forgot to pass the
dataset to the new form
        If (f.ShowDialog() = Windows.Forms.DialogResult.OK) Then
            savetable = True
            DataSetClub.AcceptChanges()
        End If

    End Sub

    Private Sub ButtonRecordTrainingSession_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ButtonRecordTrainingSession.Click

        Dim f As New Form_Training_Session

        f.ds = DataSetClub

        If (f.ShowDialog() = DialogResult.OK) Then
            DataSetClub.Tables("TableTrainingSession").Rows.Add(Nothing,
            f.ComboBoxMembers.SelectedValue,
f.ComboBoxActivity.SelectedValue,
            f.TextBoxDuration.Text, f.TextBoxDistance.Text,
            f.DateTimePickerSessionDate.Value.Date, f.TextBoxCalories.Text,
```

```
f.TextBoxPoints.Text)
            savetable = True

            DataSetClub.AcceptChanges()

        End If

    End Sub

    Private Sub ButtonByWeek_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ButtonByWeek.Click
        Dim f As New Form_Summary
        Dim drv As DataRowView = ComboBoxMembers.SelectedItem
        f.Text = "Summary for " + drv("FirstName") + " " + drv("SecondName")
        f.ds = DataSetClub
        f.memberid = ComboBoxMembers.SelectedValue
        f.startd = startDate
        f.ShowDialog()

    End Sub


    Private Sub ButtonRank_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ButtonRank.Click

        Dim f As New FormRank

        f.days = NumericUpDownRank.Value
        f.ds = DataSetClub
        f.ShowDialog()

    End Sub


    Private Sub ButtonDays_Click(ByVal sender As System.Object, ByVal e
```

```
As System.EventArgs) Handles ButtonDays.Click

        Dim f As New FormDisplayTraining
        Dim drv As DataRowView = ComboBoxMembers.SelectedItem
        f.Text = "Training record of " & drv("FirstName") & " " &
drv("SecondName")
        f.days = NumericUpDownDays.Value
        f.ds = DataSetClub
        f.memberID = ComboBoxMembers.SelectedValue
        f.LabelFirstName.Text = drv("FirstName")
        f.ShowDialog()

    End Sub

    Private Sub ButtonView_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ButtonView.Click
        Dim f As New FormDisplayTraining
        Dim drv As DataRowView = ComboBoxMembers.SelectedItem
        f.Text = "Training record of " & drv("FirstName") & " " &
        drv("SecondName")
        f.LabelFirstName.Text = drv("FirstName")
        f.ds = DataSetClub
        f.memberID = ComboBoxMembers.SelectedValue
        f.ShowDialog()
    End Sub


End Class
```

NewRunner

```
Public Class NewRunner

    Friend ds As DataSet

    Private Sub ButtonAddMember_Click(sender As Object, e As EventArgs)
Handles ButtonAddMember.Click

        ds.Tables("TableMembers").Rows.Add(Nothing,
TextBoxFirstName.Text, TextBoxSurname.Text,
DateTimePickerDOB.Value, TextBoxAddress.Text, TextBoxCity.Text,
TextBoxPostCode.Text, TextBoxContact.Text, TextBoxGender.Text,
TextBoxEmailAddress.Text)

        Me.DialogResult = DialogResult.OK

    End Sub

    Private Sub ButtonCancel_Click(sender As Object, e As EventArgs)
Handles ButtonCancel.Click

        Me.DialogResult = DialogResult.Cancel

    End Sub

    Private Sub TextBoxSurname_TextChanged(sender As Object, e As
EventArgs) Handles TextBoxSurname.TextChanged

    End Sub
End Class
```

Form_Training_Session

```
Public Class Form_Training_Session
    Friend ds As DataSet
    Private loaded As Boolean = False



    Private Function calculatecalories(ByVal activityid As Integer, ByVal
minutes As Integer, ByVal metres As Integer) As Integer

        Dim result As Integer = 0
        Dim speedmph As Decimal

        Dim cph As Integer = 0

        If (activityid = 0) Then 'running

            speedmph = (TextBoxDistance.Text / TextBoxDuration.Text) *
0.0372822715    'mph

            If (speedmph <= 5) Then

                cph = 472

            ElseIf (speedmph < 6) Then

                cph = 590

            ElseIf (speedmph < 7) Then

                cph = 679

            ElseIf (speedmph < 8) Then

                cph = 797

            ElseIf (speedmph < 9) Then
```

```
            cph = 885
        Else
            cph = 944
        End If
        result = cph * (TextBoxDuration.Text / 60)
        Return result
    End If
    If (activityid = 1) Then 'cycling
        speedmph = (TextBoxDistance.Text / TextBoxDuration.Text) *
0.0372822715    'mph
        If (speedmph < 10) Then
            cph = 236
        ElseIf (speedmph < 12) Then
            cph = 354
        ElseIf (speedmph < 14) Then
            cph = 472
        ElseIf (speedmph < 16) Then
            cph = 590
```

```
        ElseIf (speedmph < 20) Then
            cph = 708
        Else
            cph = 944
        End If
        result = cph * (TextBoxDuration.Text / 60)
        Return result
    End If
    Dim hours As Decimal = TextBoxDuration.Text / 60
    If (activityid = 2) Then 'front crawl slow
        Return hours * 413
    End If
    If (activityid = 3) Then 'front crawl fast
        Return hours * 590
    End If
    If (activityid = 4) Then 'backstroke
        Return hours * 413
    End If
```

```
        If (activityid = 5) Then 'breaststroke
            Return hours * 590
        End If
        If (activityid = 6) Then 'butterfly
            Return hours * 649
        End If
        Return result
    End Function
    Private Function calculatepoints(ByVal activityid As Integer, ByVal
calories As Integer, ByVal sessionTime As Integer, ByVal sessiondistance
As Integer) As Decimal
        Dim result As Decimal = 0
        result = result + calories / 500
        If (activityid = 0) Then
            If sessionTime >= 60 Then
                Dim mph As Decimal = (sessiondistance / sessionTime) *
0.0372822715
                If mph > 6 Then result = result + (mph - 6) / 6
            End If
        End If
```

```
        Return result

    End Function

    Private Sub ButtonOK_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ButtonOK.Click

        ' Make sure that the data is valid before making the calculations

        Dim ValidData As Boolean = True

        If DateTimePickerSessionDate.Value.Date > Date.Today Or
DateTimePickerSessionDate.Value.Date < #10/1/2014# Then

            MessageBox.Show("Invalid Date")

            ValidData = False

        End If

        If TextBoxDistance.Text = "0" Then

            MessageBox.Show("Invalid Distance")

            ValidData = False

        End If

        If TextBoxDuration.Text = "0" Then

            MessageBox.Show("Invalid Duration")

            ValidData = False

        End If
```

```
        If ValidData Then

            Me.DialogResult = DialogResult.OK

        End If

    End Sub

    Private Sub ButtonCancel_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ButtonCancel.Click

        Me.DialogResult = DialogResult.Cancel

    End Sub

    Private Sub FormTrainingSession_FormClosing(ByVal sender As
System.Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles
MyBase.FormClosing

        'The Calories and Points textboxes are filled in on form closing, and
could well be invisible

        If Me.DialogResult = DialogResult.OK Then
            TextBoxCalories.Text =
calculatecalories(ComboBoxActivity.SelectedValue,TextBoxDuration.Text,
TextBoxDistance.Text)
            TextBoxPoints.Text =
calculatepoints(ComboBoxActivity.SelectedValue,
            TextBoxCalories.Text, TextBoxDuration.Text,
TextBoxDistance.Text)

        End If
```

```
    End Sub


    Private Sub Form_Training_Session_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
        ComboBoxMembers.DataSource = ds.Tables("TableMembers")
        ComboBoxMembers.DisplayMember = "FirstName"
        ComboBoxMembers.ValueMember = "ID"

        ComboBoxActivity.DataSource = ds.Tables("TableSports")
        ComboBoxActivity.DisplayMember = "Description"
        ComboBoxActivity.ValueMember = "ID"

        loaded = True

    End Sub
End Class
```

FormDisplayTraining

```
Public Class FormDisplayTraining

    Friend ds As DataSet
    Friend memberID As Integer
    Friend days As Integer = 0
    Private bs As BindingSource

    Private Function calculatecalories(ByVal activityid As Integer) As Integer
        Dim result As Integer = 0

        Dim drv As DataRowView


        For Each o As Object In bs.List
```

```
            drv = o
            If (drv("sportID") = activityid) Then result = result +
drv("CalorificValue")
        Next
        Return result
    End Function

    Private Function calculatepoints() As Decimal
        Dim result As Decimal = 0

        Dim drv As DataRowView


        For Each o As Object In bs.List
            drv = o
            result = result + drv("points")
        Next
        Return result
    End Function

    Private Sub FormDisplayTraining_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load

        bs = New BindingSource
        bs.DataSource = ds.Tables("TableTrainingSession")
        bs.Filter = "memberID = " & memberID

        Dim d As Date = Date.Today
        Dim dstr As String

        If days > 0 Then

            d = d.AddDays(-days)
            dstr = d.Month & "/" & d.Day & "/" & d.Year
```

```
            bs.Filter += " AND  SessionDate >= #" & dstr & "#"

        End If

        ' DataGridViewMySesssions.DataSource = bs Comments from Negem:
the name of the DataGrid is wrong Sesssions
        DataGridViewMySessions.DataSource = bs

        TextBoxRunningK.Text = calculatecalories(0)
        TextBoxCyclingK.Text = calculatecalories(1)

        TextBoxSwimmingK.Text = calculatecalories(2) + calculatecalories(3)
+ calculatecalories(4) +
        calculatecalories(5) + calculatecalories(6)

        TextBoxCalories.Text = calculatecalories(0) + calculatecalories(1) +
calculatecalories(2) +
        calculatecalories(3) + calculatecalories(4) + calculatecalories(5) +
calculatecalories(6)

        TextBoxPoints.Text = Format(calculatepoints(), "N")
        TextBoxAvRunning.Text = Format(calculateAverage(0, days), "N")
        TextBoxAvCycling.Text = Format(calculateAverage(1, days), "N")
        TextBoxAvSwimming.Text = Format(calculateAverageSwim(days), "N")
        TextBoxAvTotalCalories.Text = Format(calculateAvAllActivities(days),
"N")
        TextBoxAveragePoints.Text = Format(calculateAvPoints(days), "N")

    End Sub

    Private Function calculateAverage(ByVal activityid As Integer, ByVal
days As Integer) As Decimal

        Dim d As Date = Date.Today
```

```
        d = d.AddDays(-days)

        Dim tot As Decimal = 0

        Dim l As New ArrayList

        For Each dr As DataRow In ds.Tables("TableTrainingSession").Rows

            If dr("memberID") <> memberID And dr("sportid") = activityid And
(days =
            0 Or dr("SessionDate") >= d) Then

                tot = tot + dr("CalorificValue")
                If Not l.Contains(dr("memberid")) Then l.Add(dr("memberid"))

            End If

        Next

        If (l.Count <> 0) Then Return tot / l.Count

        Return 0

    End Function

    Private Function calculateAverageSwim(ByVal days As Integer) As
Decimal

        Dim d As Date = Date.Today

        d = d.AddDays(-days)

        Dim tot As Decimal = 0

        Dim l As New ArrayList
```

```
        For Each dr As DataRow In ds.Tables("TableTrainingSession").Rows

            If dr("memberID") <> memberID And dr("sportid") >= 2 And (days =
0 Or
            dr("SessionDate") >= d) Then

                tot = tot + dr("CalorificValue")
                If Not l.Contains(dr("memberid")) Then l.Add(dr("memberid"))

            End If

        Next

        If (l.Count <> 0) Then Return tot / l.Count

        Return 0

    End Function

    Private Function calculateAvAllActivities(ByVal days As Integer) As
Decimal

        Dim d As Date = Date.Today

        d = d.AddDays(-days)

        Dim tot As Decimal = 0

        Dim l As New ArrayList

        For Each dr As DataRow In ds.Tables("TableTrainingSession").Rows

            If dr("memberID") <> memberID And (days = 0 Or
dr("SessionDate") >= d) Then

                tot = tot + dr("CalorificValue")
```

```
            If Not l.Contains(dr("memberid")) Then l.Add(dr("memberid"))
        End If
    Next
    If (l.Count <> 0) Then Return tot / l.Count
    Return 0
End Function
Private Function calculateAvPoints(ByVal days As Integer) As Decimal
    Dim d As Date = Date.Today
    d = d.AddDays(-days)
    Dim tot As Decimal = 0
    Dim l As New ArrayList
    For Each dr As DataRow In ds.Tables("TableTrainingSession").Rows
        If dr("memberID") <> memberID And (days = 0 Or
dr("SessionDate") >= d) Then
            tot = tot + dr("Points")
            If Not l.Contains(dr("memberid")) Then l.Add(dr("memberid"))
        End If
    Next
    If (l.Count <> 0) Then Return tot / l.Count
```

```
        Return 0

    End Function

    Private Sub DataGridViewMySessions_CellContentClick(sender As
Object, e As DataGridViewCellEventArgs) Handles
DataGridViewMySessions.CellContentClick

    End Sub
End Class
```

FormSummary

```
Public Class Form_Summary


    Friend ds As DataSet
    Friend memberid As Integer
    Friend startd As Date
    Private bs As BindingSource

    Private Function calccals() As Integer
        Dim result As Integer = 0
        Dim drv As DataRowView
        For Each o As Object In bs.List
            drv = o
            result = result + drv("CalorificValue")
        Next
        Return result
    End Function

    Private Function calcpoints() As Decimal
        Dim result As Decimal = 0
        Dim drv As DataRowView
```

```
        For Each o As Object In bs.List
            drv = o
            result = result + drv("points")
        Next
        Return result
    End Function

    Private Sub FormSummary_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load

        bs = New BindingSource
        bs.DataSource = ds.Tables("TableTrainingSession")
        Dim done As Boolean = False
        Dim d As Date = startd
        Dim weeknumber As Integer = 0
        Dim totalCalories As Integer = 0
        Dim totalPoints As Decimal = 0.0
        Do While Not done
            weeknumber = weeknumber + 1
            bs.Filter = "memberID = " & memberid
            Dim d1 As Date = d
            Dim dstr1 As String = d1.Month & "/" & d1.Day & "/" & d1.Year
            d = d.AddDays(7)
            Dim dst As String = d.Month & "/" & d.Day & "/" & d.Year
            If d >= Date.Today Then done = True
            Try
                bs.Filter += " AND SessionDate >= #" & dstr1 & "# AND
SessionDate <= #" & dst & "#"
                DataSetSummary.Tables(0).Rows.Add(weeknumber, calccals,
calcpoints)
                totalCalories = totalCalories + calccals()
                totalPoints = totalPoints + calcpoints()
            Catch ex As Exception
```

```
            MessageBox.Show(ex.Message & " " & bs.Filter)
        End Try
    Loop
    DataGridViewSummary.DataSource = DataSetSummary.Tables(0)
    LabelTotalCalories.Text += " " & totalCalories
    LabelTotalPoints.Text += " " & Format(totalPoints, "N")

End Sub


End Class
```

FormRank

```
Public Class FormRank

    Friend ds As DataSet
    Friend days As Integer = 0

    Private Function points(ByVal days As Integer, ByVal memberID As
Integer) As Double

        Dim bs As New BindingSource
        bs.DataSource = ds.Tables("TableTrainingSession")
        bs.Filter = "memberID = " & memberID

        Dim d As Date = Date.Today
        Dim dstr As String

        d = d.AddDays(-days)
        dstr = d.Month & "/" & d.Day & "/" & d.Year

        bs.Filter += " AND  SessionDate >= #" & dstr & "#"
        Dim result As Double = 0
```

```
        Dim i As Integer
        For i = 0 To 6
            result = result + calculatepoints(i, bs)
        Next
        Return result

    End Function

    Private Function calculatepoints(ByVal activityid As Integer, ByVal bs As
BindingSource) As Double

        Dim result As Double = 0
        Dim drv As DataRowView

        For Each o As Object In bs.List
            drv = o
            If (drv("sportID") = activityid) Then result = result +
            drv("CalorificValue") / 500
            If (drv("sportID") = 0) Then
                If drv("SessionTime") >= 60 Then
                    Dim mph As Double = drv("SessionDistance") /
drv("SessionTime") * 0.0372822715
                    If mph > 6 Then result = result + (mph - 6) / 6
                End If
            End If
        Next
        Return result

    End Function

    Private Sub FormRank_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load

        Dim bs As New BindingSource
```

```
        bs.DataSource = ds.Tables("TableTrainingSession")
        For Each dr As DataRow In ds.Tables("TableMembers").Rows
            If DataSetResults.Tables(0).Rows Is Nothing Then
MessageBox.Show("null")
            Try
                DataSetResults.Tables(0).Rows.Add(dr("ID"), dr("FirstName"),
                dr("SecondName"), points(days, dr("ID")))
            Catch ex As Exception
                MessageBox.Show(ex.Message)
            End Try
        Next
        DataSetResults.AcceptChanges()
        DataGridViewResults.DataSource = DataSetResults.Tables(0)
        DataGridViewResults.Sort(DataGridViewResults.Columns("Points"),
        System.ComponentModel.ListSortDirection.Descending)

    End Sub

    Private Sub DataGridViewResults_CellContentClick(sender As Object, e
As DataGridViewCellEventArgs) Handles
DataGridViewResults.CellContentClick

    End Sub
End Class
```

Parkwood Vale Harriers
Running club

Testing

&

Evaluation

Testing & Evaluation

Test Strategy

There are two stages to my testing; alpha testing and beta testing.

Alpha testing will be a simulated form of testing, I will do this myself. The alpha testing will consist of something called white box testing. This is where the testing is done by someone who is aware of how the program is supposed to work. This is so that tests can be designed appropriately to test how and if the program is working as it should.

I will be testing each form and taking into consideration the way in which they work. This should enable me to determine whether these forms are working correctly.

I will carry these tests out and document whether each form worked as I had expected or not. This will be very useful when evaluating the success of the program.

Beta testing is going to be done by an end user. This part will involve black box testing. This is where the testing strategy is performed by an individual who does not have an experience in how the system was programmed and what language was used. I will design tasks for the tester to carry out on various forms. Once the task list is completed I will give the tester a questionnaire so that I can gain feedback on their experiences with the program. This is also going to be very useful when I create my final evaluation of the success of the program.

Alpha Testing

Test Data

This test data will be from my alpha testing stage. In this section all buttons and commands will be tested thoroughly to ensure we have a good grasp on the functionality of the program.

No	Form	Test Data	Reason	Expected Result	Actual Result
1	TrainingLog	N/A	Testing that	I expect that once	See

			the Add New Member button works correctly.	this button is selected I will be taken to the NewRunner form.	Print Screen. 1
2	NewRunner	First name: Ryan Surname: Jones DOB: 10/2/1995 Address: 57 North Walk City: Cardiff Postcode: CF62 8BX Contact: +44 7983 423039 Gender: Male Email: ryanjones@hotmail.co.uk	I will be testing whether or not the Add Member button is working correctly	This member will be stored on the system.	See Print Screen. 2 & 3
3	NewRunner	No data entered.	I will be testing what happens when no data is entered.	A new member will be created with no details except a date. This member will be stored on the system.	See Print Screen. 4 & 5
4	TrainingLog	N/A	I will be testing the functionality of the View Member Details button.	I will be taken to the FormMembers form.	See Print Screen. 6
5	Form Members	Change city location from ‘Cardiff’ to ‘Barry’.	Testing that when data is edited that the new and updated data is successfully stored.	Form will close itself and data will be stored.	See Print Screen. 7 & 8
6	Form Members	Edit names from ‘Ryan’ to ‘Luke’.	Testing whether or not the cancel button is working as	Form closes itself and will reject any data that was input.	See Print Screen. 9 & 10

			expected.		
7	TrainingLog	N/A	Testing whether the the 'Record Training Session' button works correctly or not.	The form Form Training Session will open.	See Print Screen. 11
8	TrainingLog	N/A	I will test to see if the Combobox actually contains all of the data it is supposed to, e.g. members' names from the dataset.	The comboox should display all 3 members from the dataset.	See Print Screen. 12
9	TrainingLog	N/A	I will test the command button 'All' to see if it works properly.	The form FormDisplayTraining should open and should show all of the data from members' training sessions.	See Print Screen. 13
10	TrainingLog	Member: Richard Days: 5	I will be testing the functionality of the 'Last Days' command button as well as ensuring the NumericUpdown is working correctly.	The form FormDisplayTraining should open and display the only the past 5 days of training sessions.	See Print Screen. 14
11	TrainingLog	Member: Richard	I will test to check the functionality of the 'Summary' command button.	Form_Summary should open and show all of the data on the selected member.	See Print Screen. 15
12	TrainingLog	Days: 30	I will be testing the functionality of the 'Rank' command button and ensuring that	FormRank should open and display all of the members' data from the past 30 days. They will be arranged in order of	See Print Screen. 16

			the NumericUpDown is working effectively.	points awarded, from highest to lowest.	
13	FormTrainingSession	Member: Ryan Activity: Running Duration: 30 Distance: 0 Date: 29th April 2015	This is designed to test the validation of the TextBoxDistance.	An error message should be displayed, reading "Invalid Distance" as the system detects a validation error.	See Print Screen. 17 & 18
14	FormTrainingSession	Member: Ryan Activity: Running Duration: 0 Distance: 5,000 Date: 29th April 2015	This is designed to test the validation of the TextBoxDuration.	An error message should be displayed, reading "Invalid Duration" as the system detects a validation error.	See Print Screen. 19 & 20
15	FormTrainingSession	Member: Ryan Activity: Running Duration: 30 Distance: 5,000 Date: 30 th April 2015	This is designed to test the validation of the DateTimePicker.	An error message should be displayed, reading "Invalid Date" as the system detects a date that is in the future, a validation error.	See Print Screen. 20 & 21.

Beta Testing

Adding New Member:

Select button 'Add New Member'

Enter:

Name: Ryan

Surname: Jones

DOB: 10/2/95

Address: North Walk

City: Cardiff

Postcode: CF87 5BX

Contact Number: 04788321697

Gender: Male

Email: ryanjones@hotmail.com

Select button ‘Add Member’

Select button ‘View Member Details’

View the new user within the DataGrid.

Editing Members’ Details

Select button ‘View Member Details’

Change member name Richard to William

Select button ‘OK’

Select button ‘View Member Details’ again

View the change in member’s name in the record.

Recording a Training Session (using valid data)

Select button ‘Record Training Session’

Select member ‘John’

Select ‘Running’

Input duration as ‘20’

Input distance as ‘5000’

Change date to ‘April 30th 2015’

Select button ‘OK’

Recording a Training Session (using invalid data)

Select button ‘Record Training Session’

Select member ‘John’

Select ‘Running’

Input duration as ‘35’

Input distance as ‘0’

Change date to ‘April 30th 2015’

Select button ‘OK’

Displaying data for all sessions taken by a specific member

Select specific member in drop down list

Select button ‘All’

View displayed data

Select button ‘OK’

Displaying a member's training sessions by selection of week.

Select a specific member in the drop down list box.

Select button 'Summary'

View data displayed

Displaying a member's training sessions from the past 3 days.

Select specific in the drop down box list

Select NumericUpDown to 3 days

Select button 'Last Days'

View the training data displayed

Viewing all members' details and their ranks

Select NumericUpDown to 10 days

Select button 'Rank'

View users' ranks

Select 'Points' to change the arrangement from highest to lowest, to lowest to highest.

Evaluation

Usability

This program must be easy to use and require little IT experience in order to navigate through the system efficiently. The usability of this application can be tested by asking people with little IT experience to use the program. When they have tried out the program I will be able to gain feedback by asking them to complete a short

questionnaire which will ask relevant questions in order to gain a user's evaluation of this computer program. I will use the following questions in the user questionnaire:

- Did you find the program easy or difficult to use?

  Very easy ☑

  Easy ☐

  Hard ☐

  Very hard ☐

- Did you need any help with any of the tasks?

  None ☑

  For some things ☐

  For most things ☐

  For all tasks ☐

  If the majority of users found it easy to use with little or no difficulty then this will be the criteria for a successful program that requires no additional modification, allowing us to evaluate the usability.

Stability

The system must be stable so that no data is lost as a result of errors. This can be fixed by testing the system thoroughly by applying logical and illogical sequences, such as entering no values in text boxes, etc. IN order to evaluate the stability I will ask users to test the system and then I will give them a

questionnaire to fill out with a simple question that should efficiently evaluate the stability. I will ask the following question to determine the stability of my system:

- Whilst using the program did you find any problems? (Errors, crashing, etc)

  Yes ☑

  No ☐

Usefulness

Perhaps most importantly the computer program has to be useful and meet the given requirements. In order to better understand the usefulness of our program I will give users the following simple question to help us determine the usefulness

- How useful do you find the computer program?

  Very useful ☑

  Quite useful ☐

  Not very useful ☐

  Not useful at all ☐

Success will be determined by a 'Quite useful' – 'Very useful' average. Given this criteria the program will be ready for actual use by Parkwood Vale Harriers.

Performance

The computer program must excel at what it is designed for. This means it must be fast and must be reliable enough for members to use this consistently on a regular basis. Users should be able to transition between forms and perform calculations with very little to no waiting time whatsoever.

In order to test the quality of performance I will give users this final questionnaire in order to evaluate the program's performance

- How much loading time did you experience whilst using the program?

  None ☑

  A little ☐

  Some ☐

  A lot ☐

Success in this field will be considered successful if users have an average answer of 'None' in this questionnaire.

Then I will consider this program to be fully functional and ready for the Parkwood Vale Harriers to begin implementation of the computer program.

Evaluation

The general brief for this project was for me to develop a system that allowed me to monitor how individuals are working. This will be used to eventually choose 8 members that are the fittest based on how many points they have earned. There must also be the ability to record training sessions and have the ability to calculate the amount of calories that are burnt and the amount of points that get earned due to the calories burnt. Overall the goal for the system is chose the 8 runners to run from John O'Groats to Land's End, this will be determined on the fittest members as the more points that a member has, the more stamina.

An essential aim for this programme is that it is easy for people with little or no IT experience to worse.

The system records data and performs simple calculations, so it must be fast and simple in operations for best usage.

This programme must be able to produce a top 10 list of runners, based on their performance. It must then display these in a ranked format, for an easy, visual way of selecting these members.

In my analysis and design work I had made evaluation criteria which would allow me to evaluate the success of this programme.

Usability

Usability is a very important aspect of my criteria was

As the system must be easy to use for people with moderate and low level IT experience, the usability is of vital importance. The usability of the system can be improved by adding easy to use command buttons and good visual indications of specific functions, e.g. text descriptions next to command controls. This will make the system much more navigable. The system must also be able to effectively perform its calculations and give the user correct data.

- Did you find the program easy or difficult to use?

Very easy ☐

Easy ☐

Hard ☐

Very hard ☐

Feedback gained from this questionnaire was all in all positive. As a result I can conclude that the usability of the system is good enough.

Performance

It is imperative that the system is up to scratch in terms of its performance. The system is reliant on the performance of itself, this is why the testing process was so important. Although we have carried out multiple black box and white box tests using valid and invalid data, it is also important that we gather information from a questionnaire, just in case we missed something.

- How much loading time did you experience whilst using the program?

  None ☐

  A little ☐

  Some ☐

  A lot ☐

Feedback showed that the speed of the system was good and that the user encountered no real issues with any performance-related issues.

Stability

The system must be stable, as if it is not stable then it will be very difficult for a user to use it properly without encountering bugs and errors. The best way for us to increase the stability of the programme is through the testing phase where a tester can provide information on the stability, by doing specific tests designed to find these stability errors (which we have designed specifically, ourselves). Though this method is thorough it is nonetheless important that we design a questionnaire for the tester to take as well.

- Whilst using the program did you find any problems? (Errors, crashing, etc)

  Yes ☐

  No ☐

Feedback showed that the tester encountered no bugs or errors given our designated tests.

Usefulness

A programme that isn't useful doesn't really have much purpose. So this stage is important so that we make a programme that is worth making and more importantly worth purchasing. The usefulness is best evaluated by users. The best method of testing usefulness of a programme is through the medium of questionnaires.

- How useful do you find the computer program?

  Very useful ☐

  Quite useful ☐

Feedback showed that users found the programme useful and that it achieved its objective of being a useful tool for members of a running club, people who regularly participate in sports and athletes alike.

Conclusion

Overall I believe the project has been successful due to the positive feedback through the testing process.

I was able to make assumptions about the system based off of the basic brief given by the club, as well as begin to think creatively about this in terms of a computer system. I was able to formulate objectives and aims off of this which helped me know what I was creating and what exactly I was implementing, and acted as a reminder of what the system is really about.

The solution was designed with the aims and objectives in mind. Then by actually creating the solution I was able to implement these aims and objectives and create a computer system that did what I had set out to make it do.

The next phase was to test this programme, to ensure it actually was doing what it was supposed to. We achieved this by means of alpha and beta testing phases and by using a both black box and white box testing to make the testing feedback that much more reliable.

The testing required designing specific tests and tasks for the testers to carry out which would give us the best form of concise feedback concerning the computer system. Tests would be designed based off of specific criteria that is designed to give the best overall feedback on every significant aspect of the computer system, such as performance or usability.

The final phase was to evaluate our feedback. Doing this allows us to have a written interpretation of the data we have gathered which is easy to access and to read. This is vital as our feedback is what is going to define this computer programme in the final stage.

The feedback we did receive from testing proved to be positive and this is precisely what allows us to consider the work successful and presentable to the client.

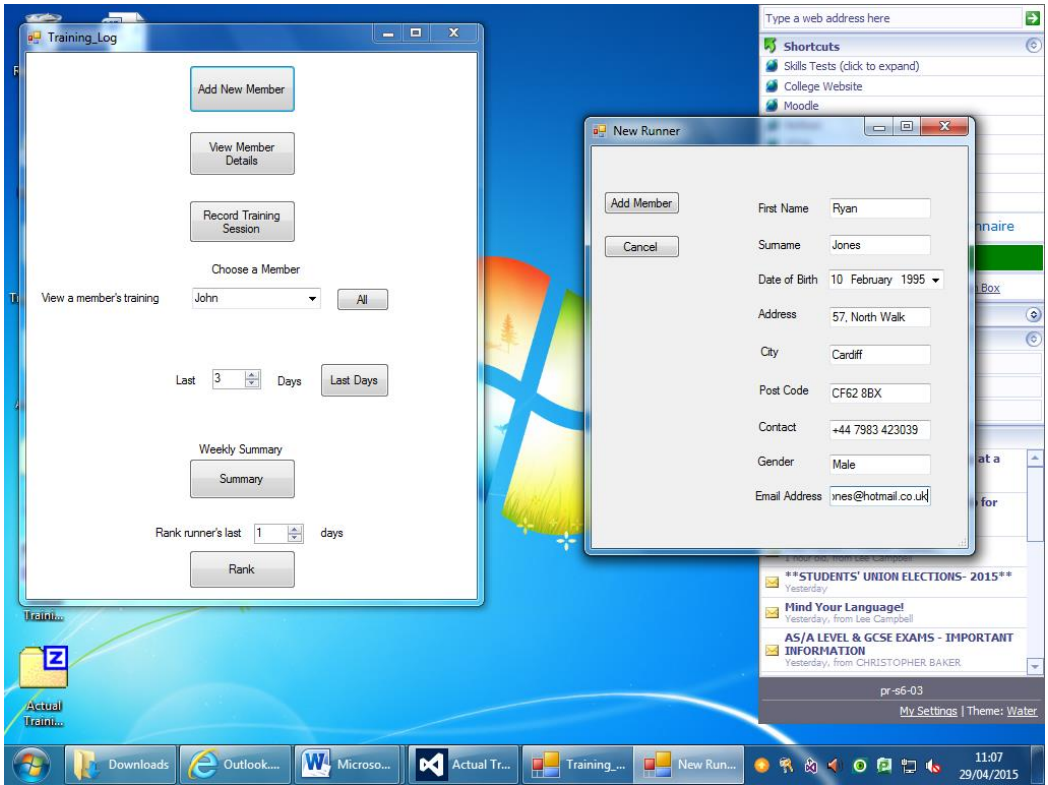
The system has managed to complete the aims and the objectives that were initially presented to us by the client, whilst also managing to add a programmer's and designer's touch to only enhance the system, such as the ability to easily view and select the top 10 members within the club.

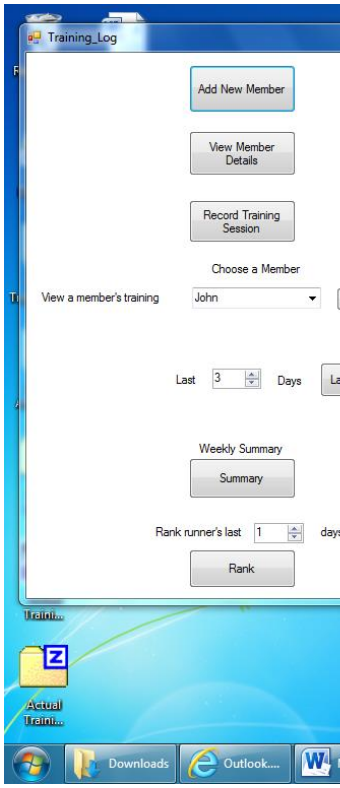
By carrying out each individual step carefully and paying close attention to the important aims and objectives, we are able to create a computer programme that should be worthy of the client's money.

By ensuring that our criteria is fulfilled and qualities such as performance, usability, stability and usefulness are successfully achieved to a high standard will allow us to feel comfortable presenting the computer application to the Parkwood Vale Harriers members and for this hopefully to be precisely what they asked us to design and create.

Print Screens

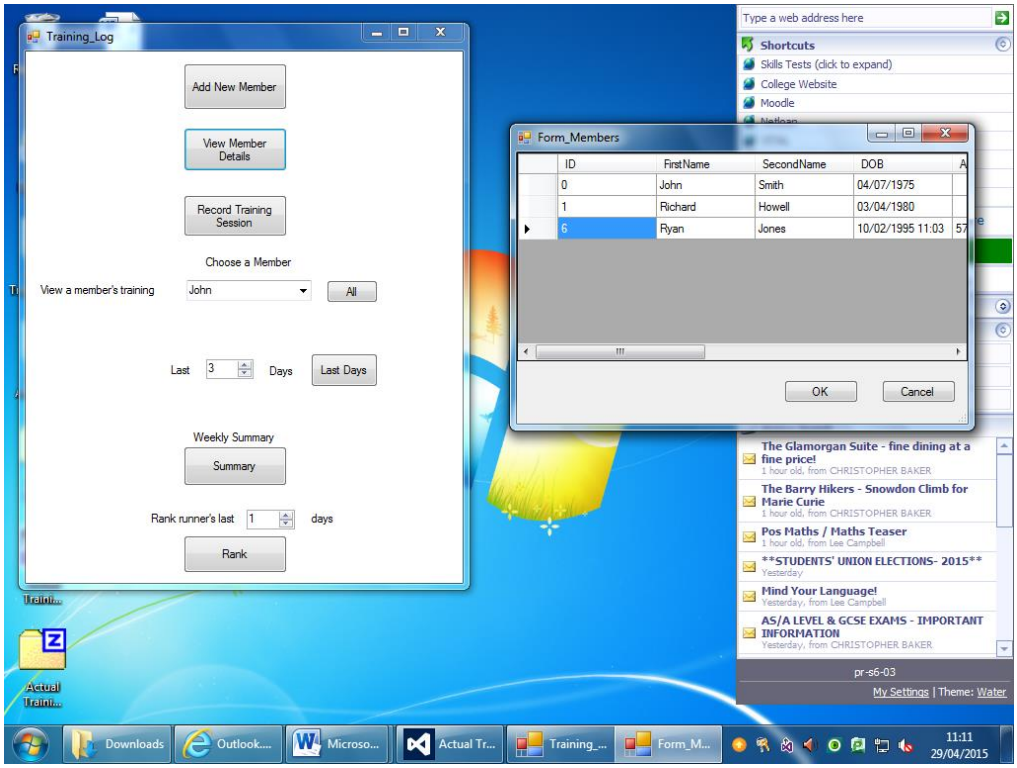
Print Screen 1.



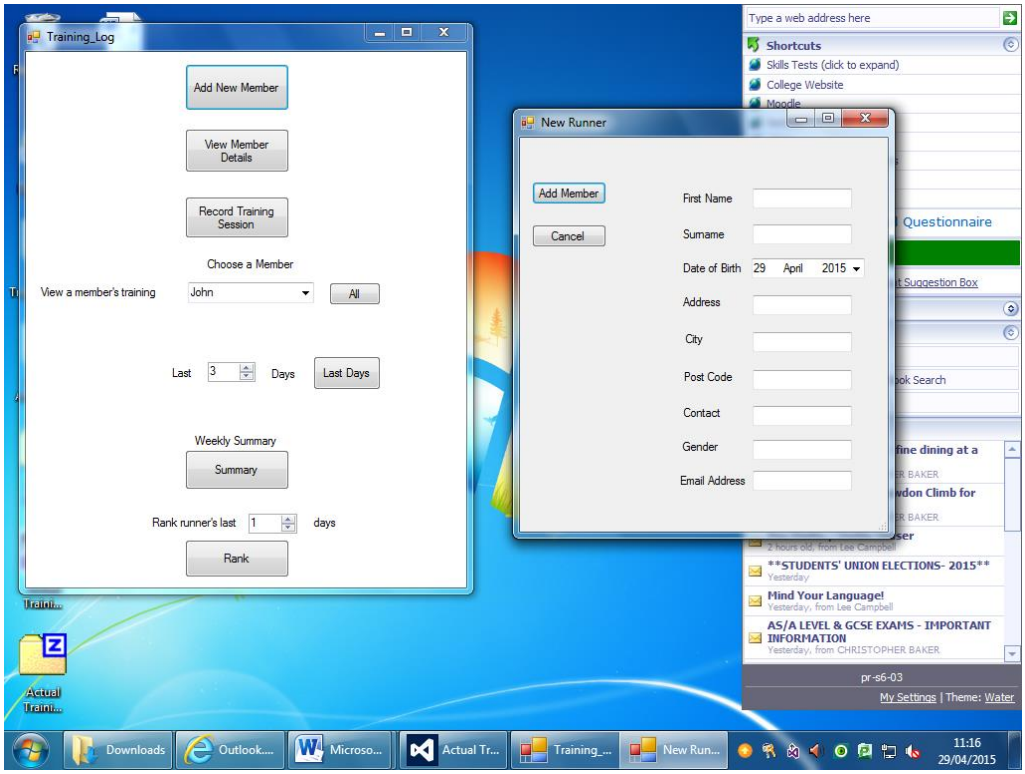


Print Screen 2.

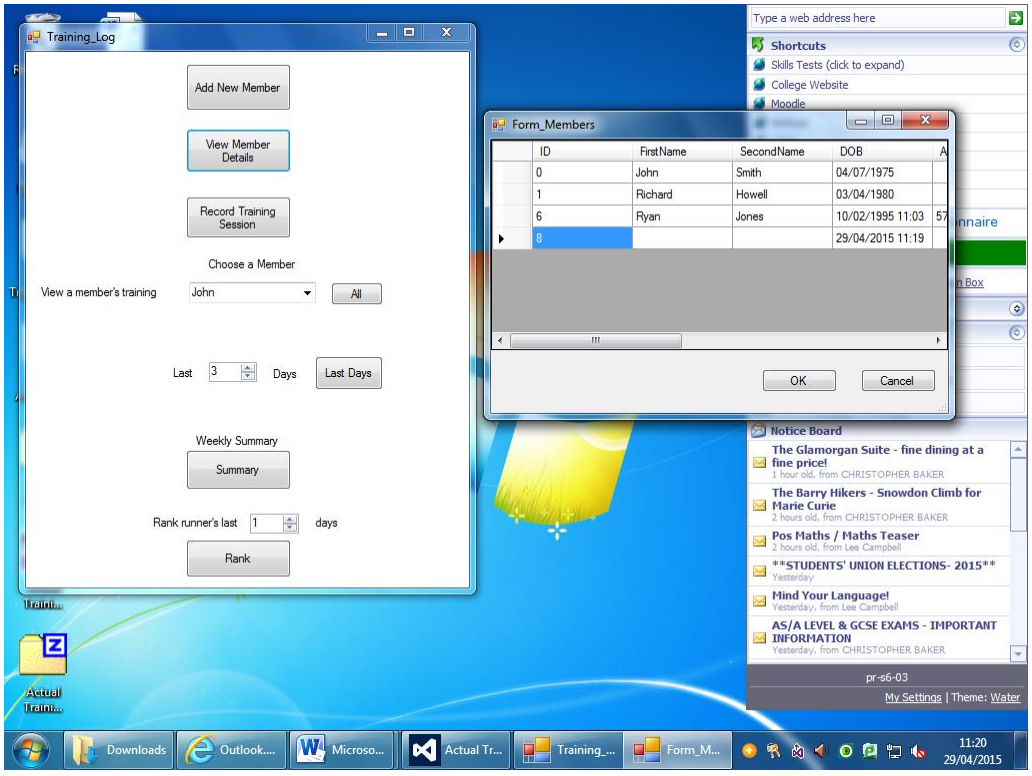
Print Screen 3.



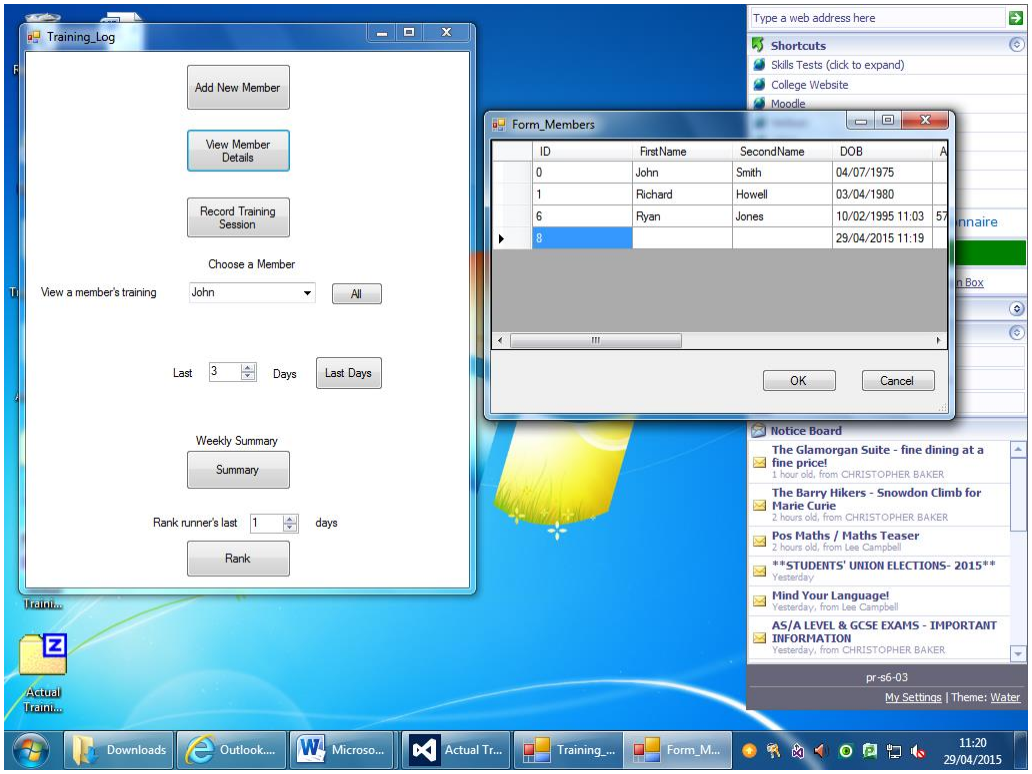
Print Screen 4.



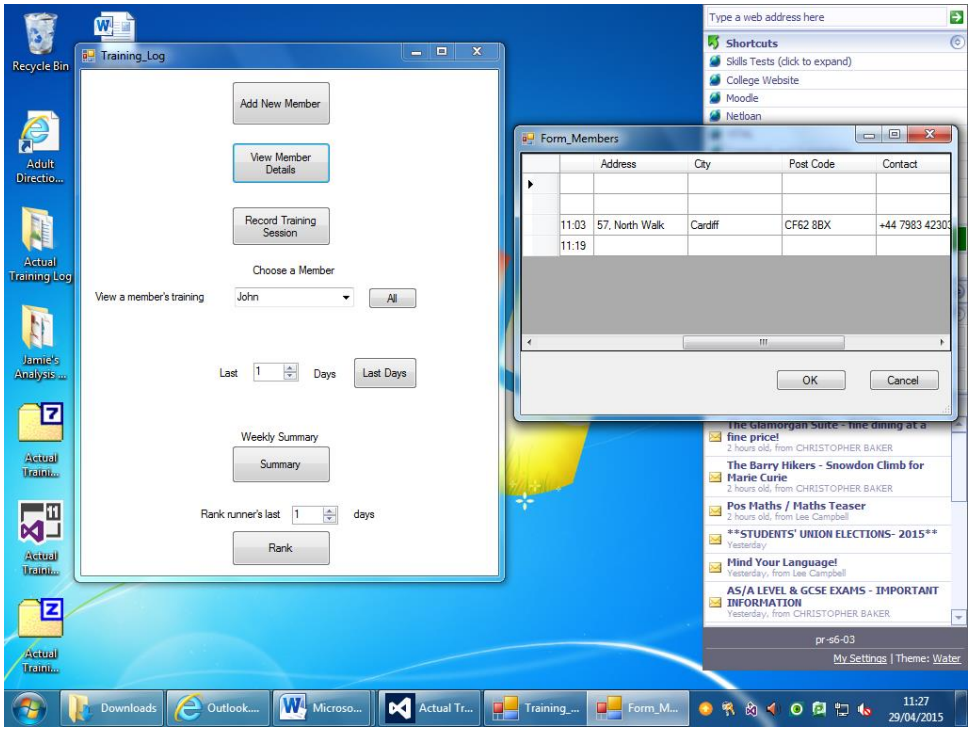
Print Screen 5.



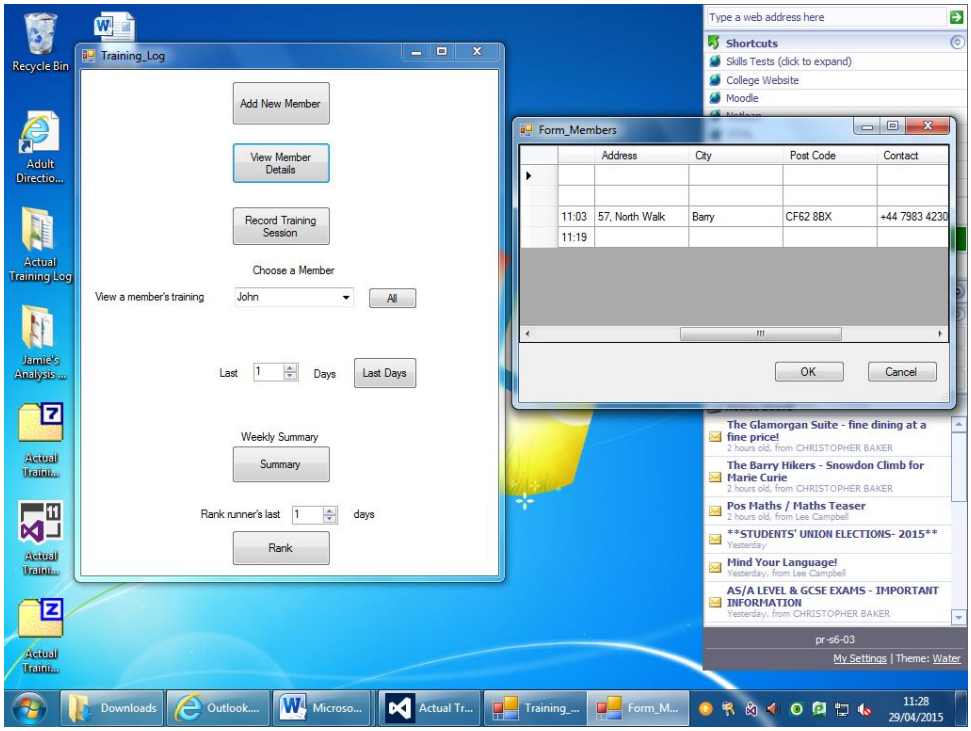
Print Screen 6.



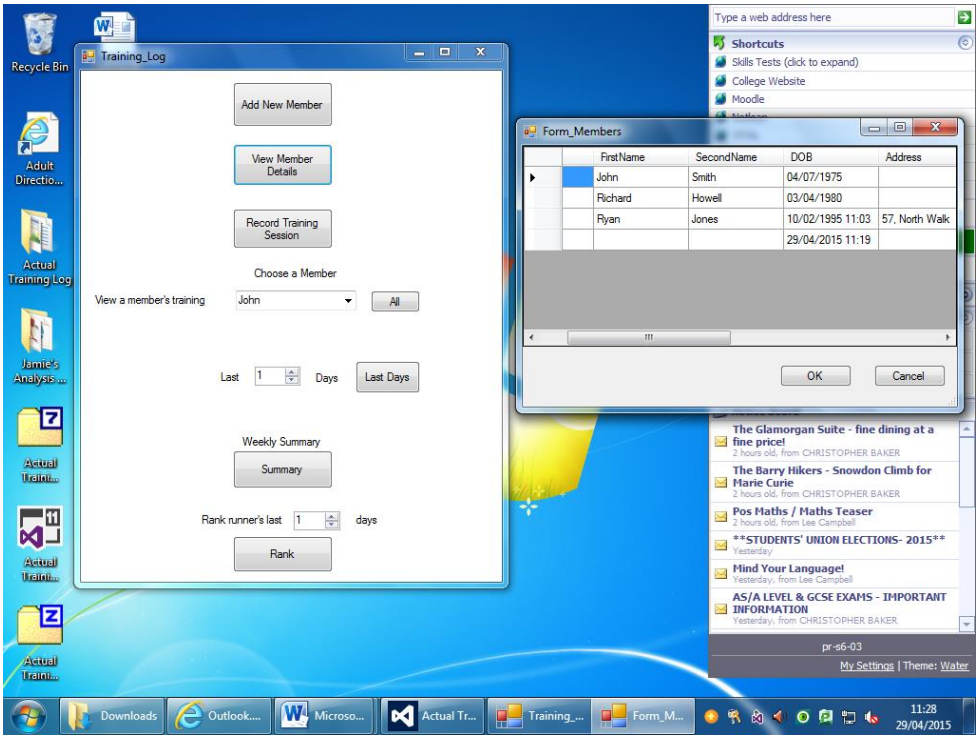
Print Screen 7.



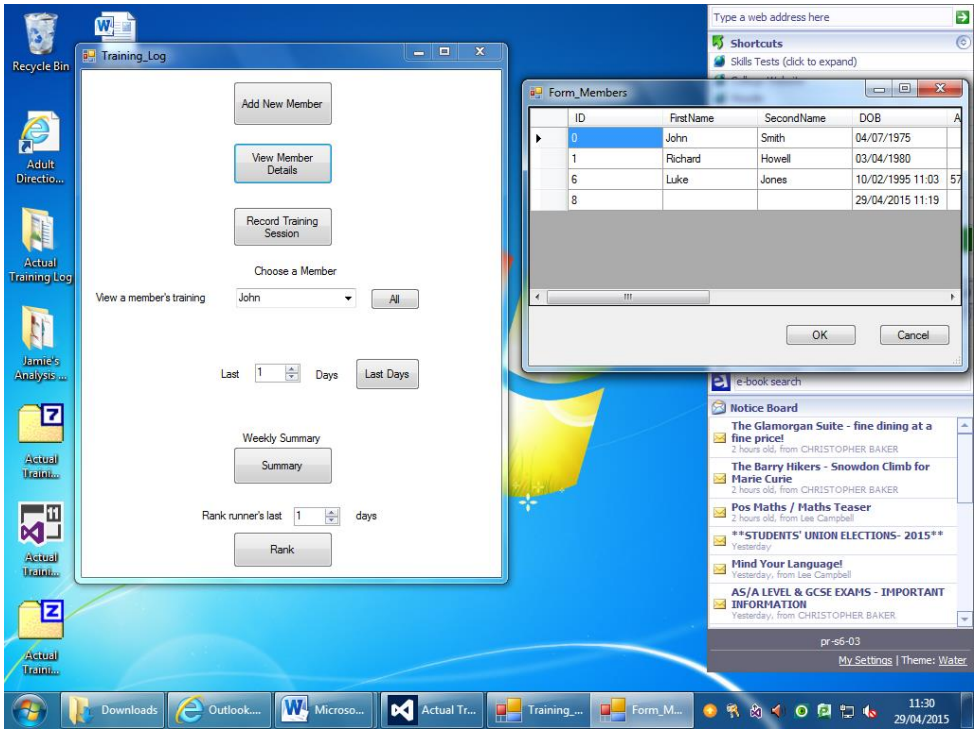
Print Screen 8.



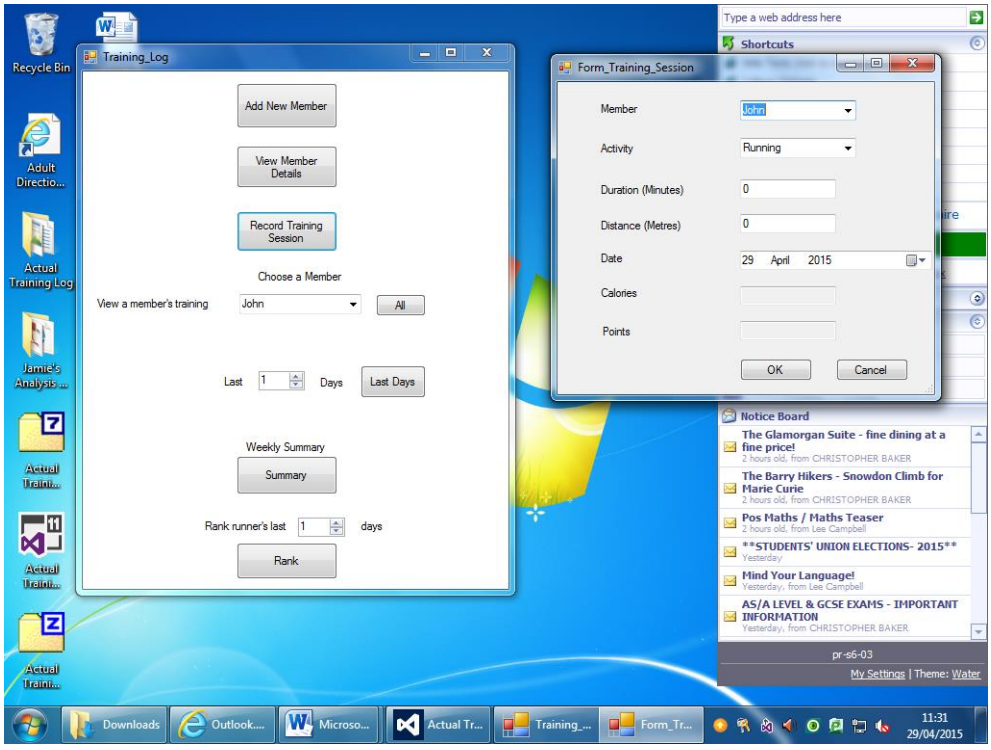
Print Screen 9.



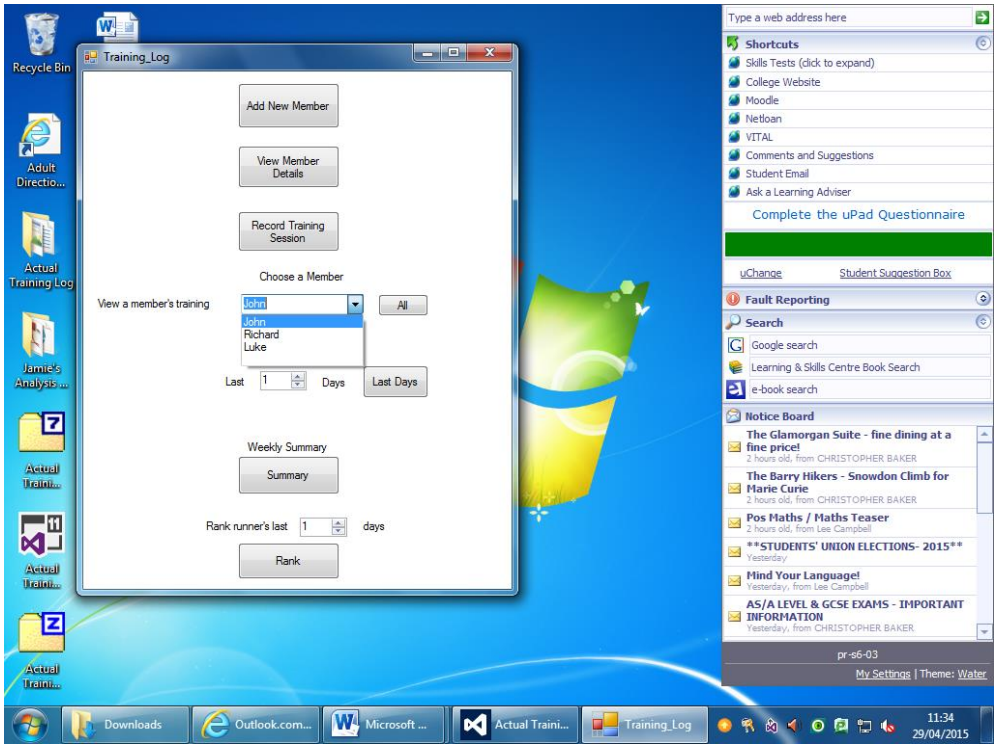
Print Screen 10.



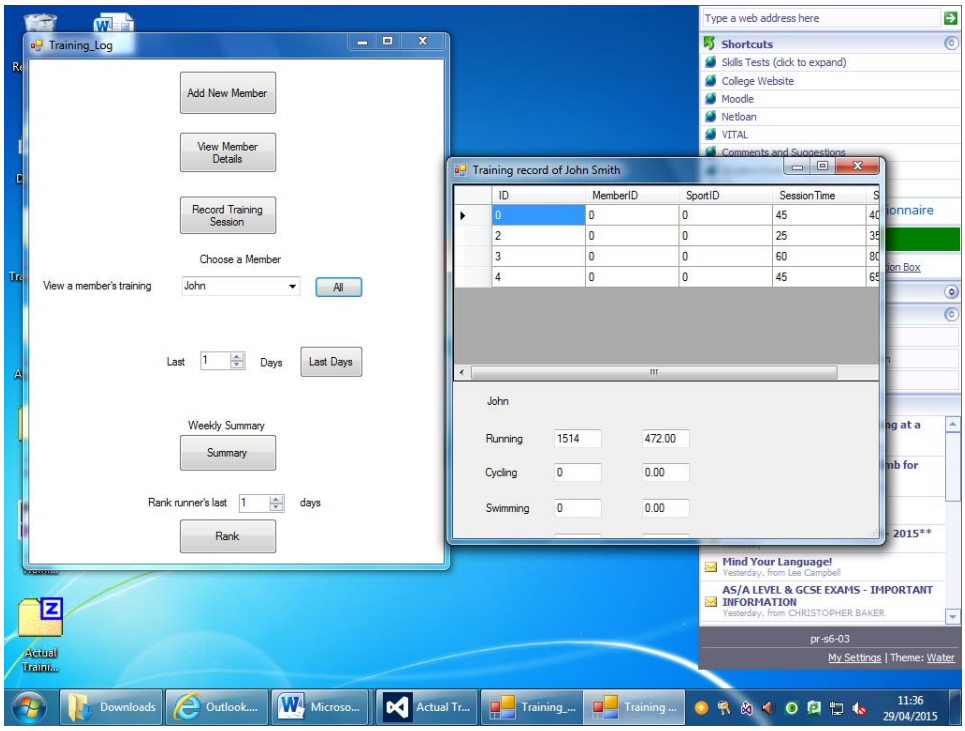
Print Screen 11.



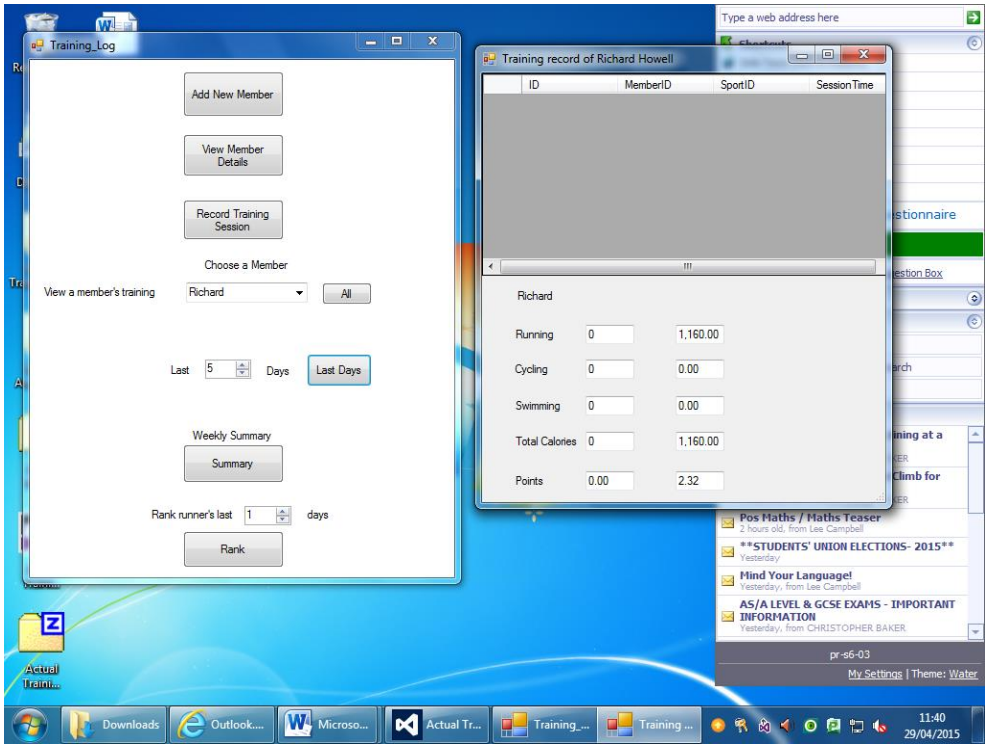
Print Screen 12.



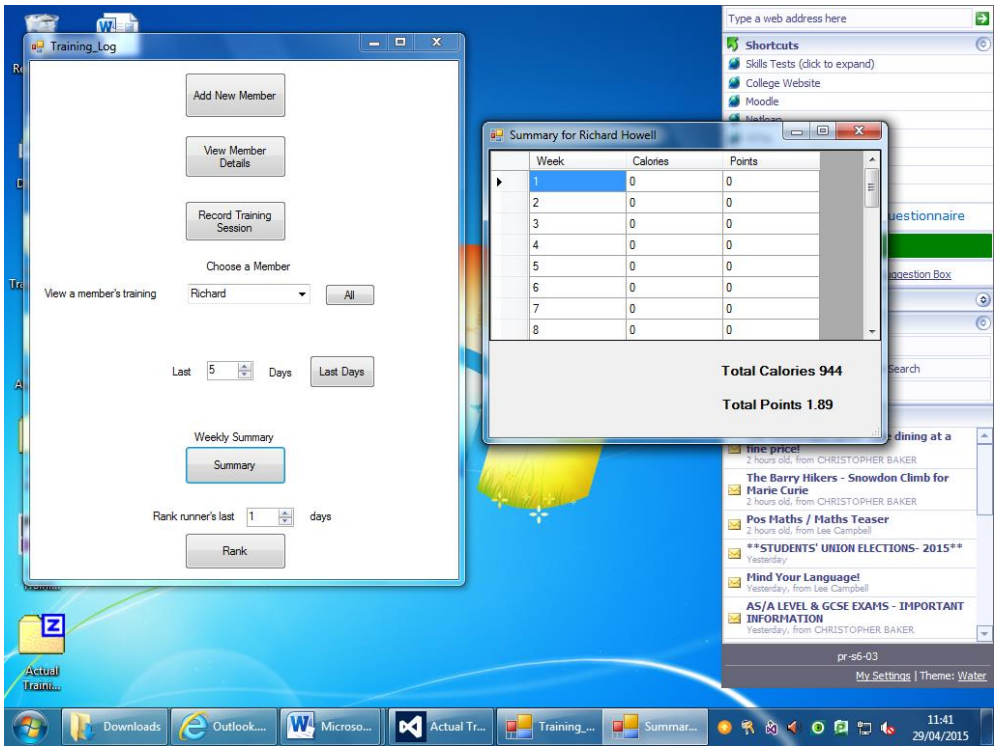
Print Screen 13.



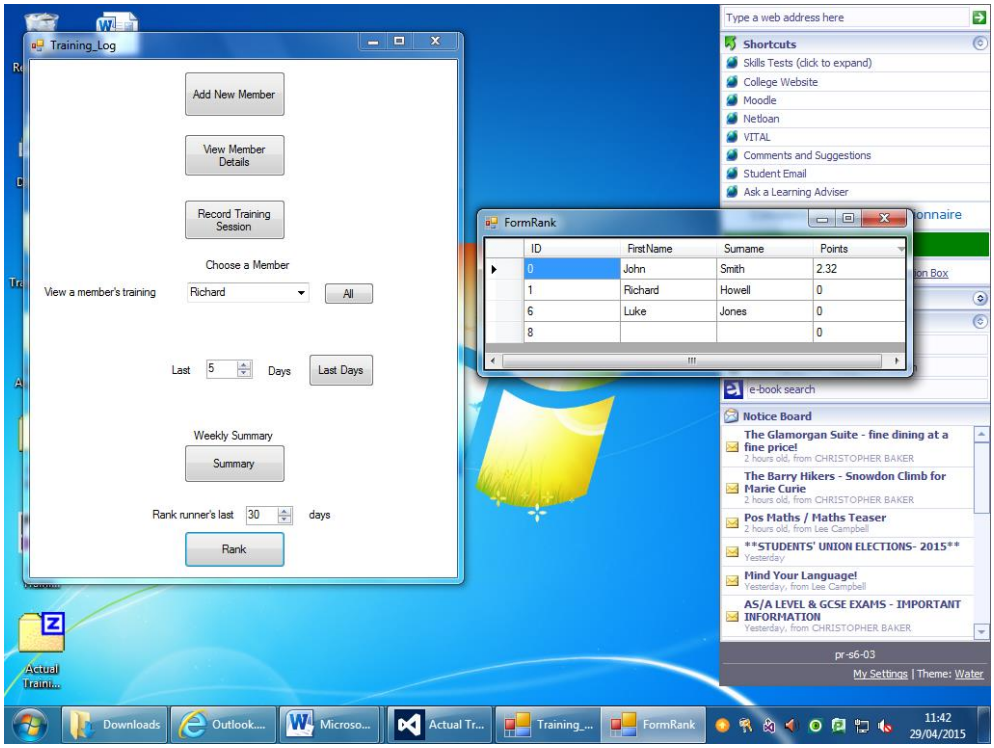
Print Screen 14.



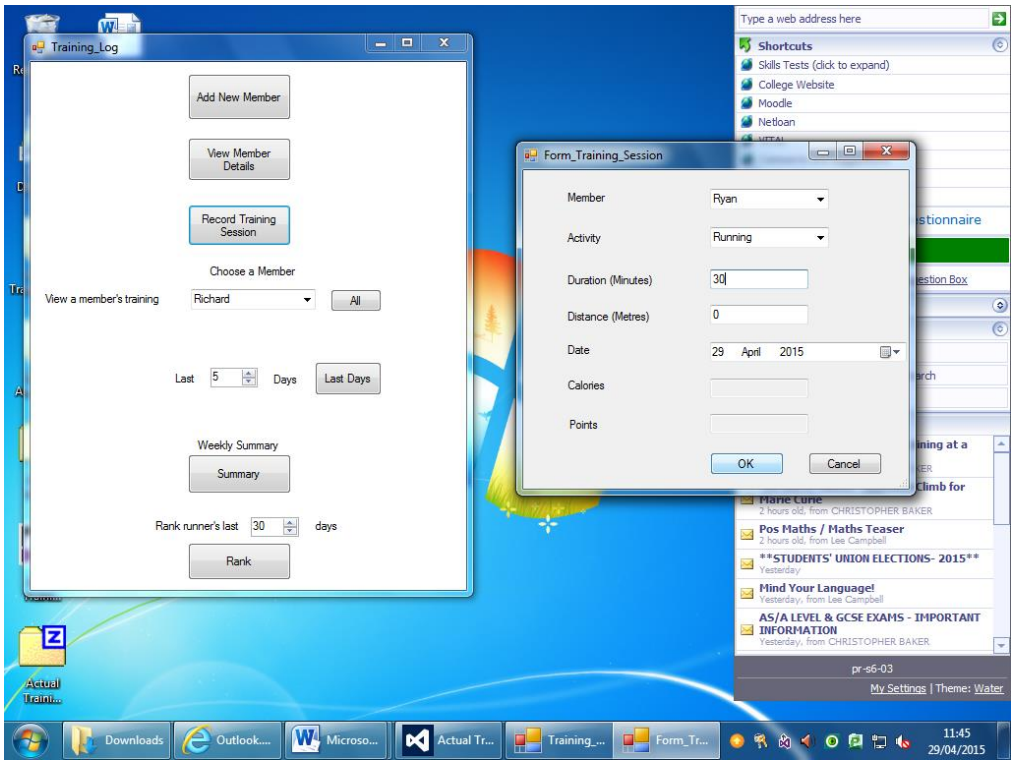
Print Screen 15.



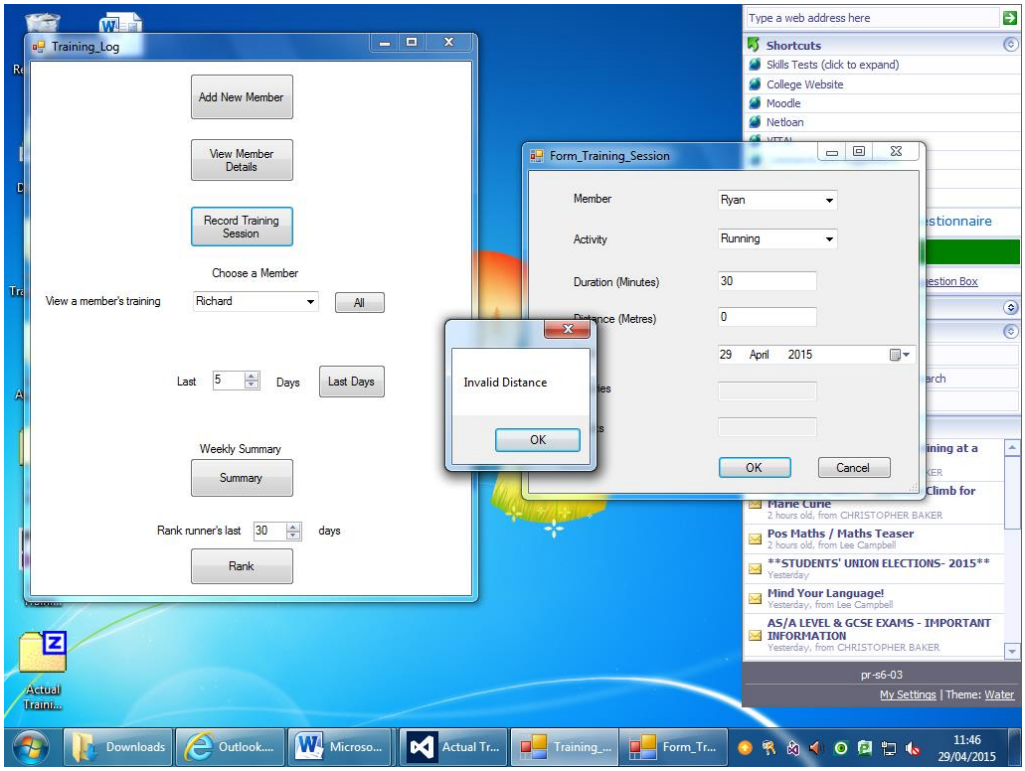
Print Screen 16.



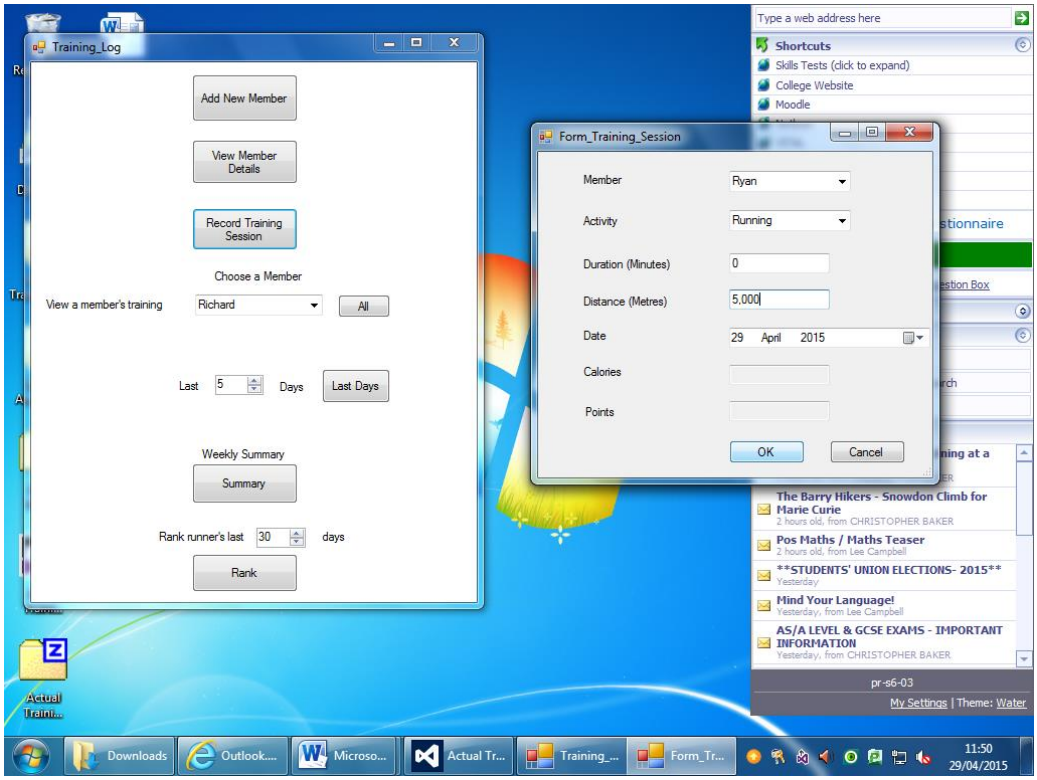
Print Screen 17.



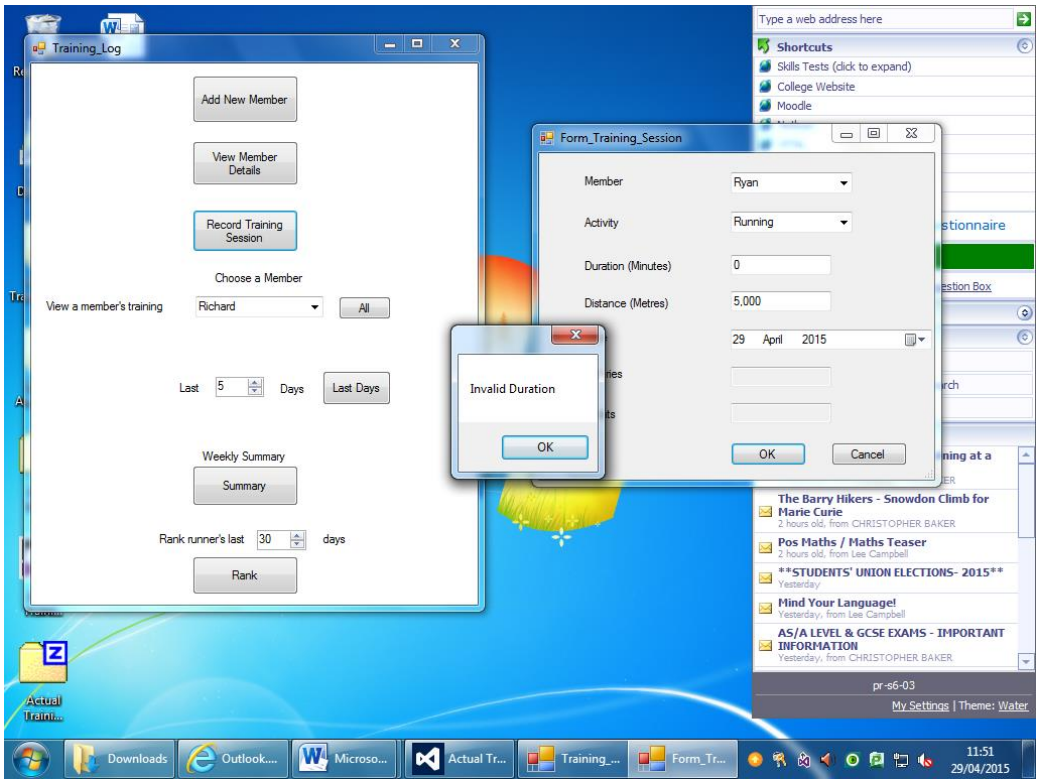
Print Screen 18.



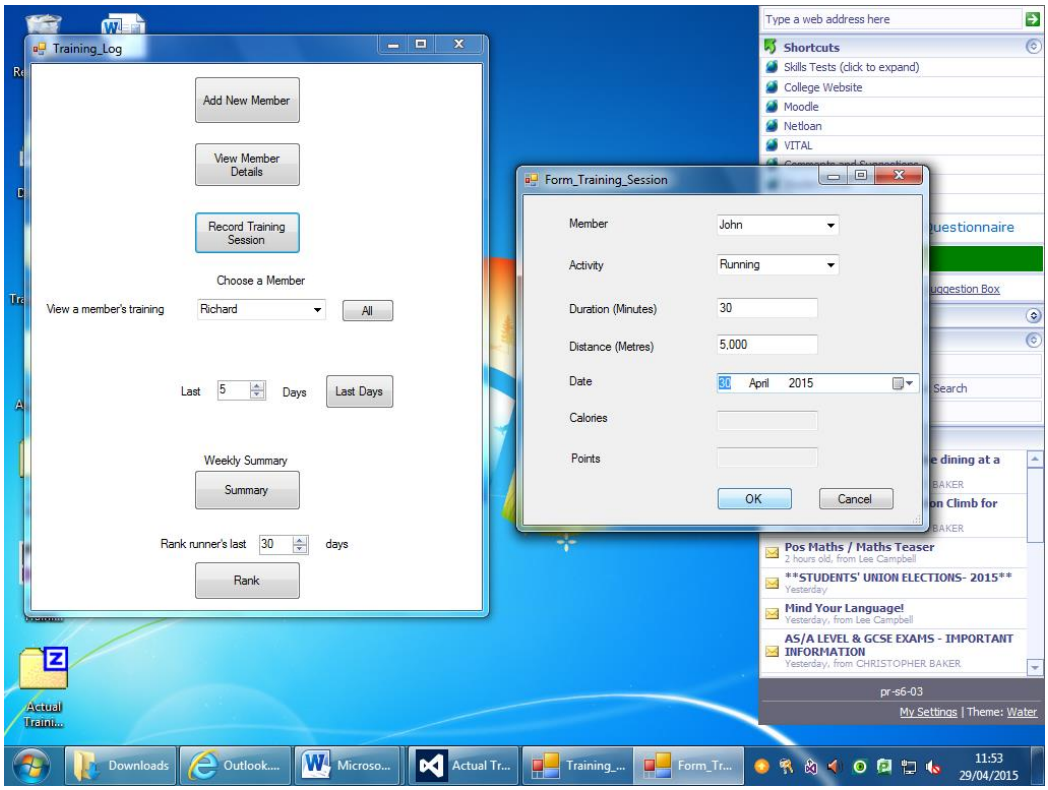
Print Screen 18.



Print Screen 19.



Print Screen 20.



Print Screen 21.