**Design**

**Problem Definition**

I am a junior employee at a small company, we recently held a lecture and the college have asked that we can design a program and judge their upcoming tournament.

**User Requirements**

The users MUST be able to enter themselves in a team or individual.

The users will be in 4 teams of 5 and 20 individuals

Each team of individual will complete 5 events.

Each event will be either team of individual.

The events will vary from sporting to academic challenges.

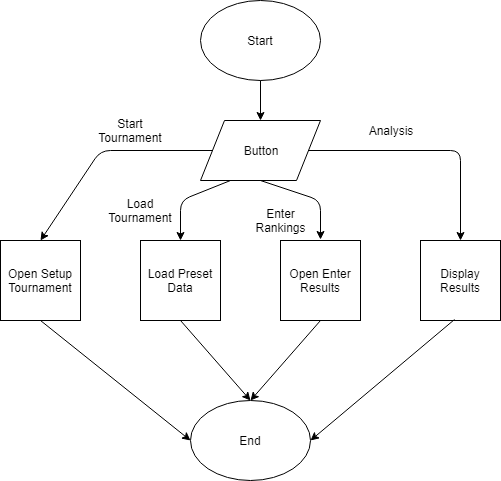
Individuals and teams will be awarded points dependant on their rank in the events.

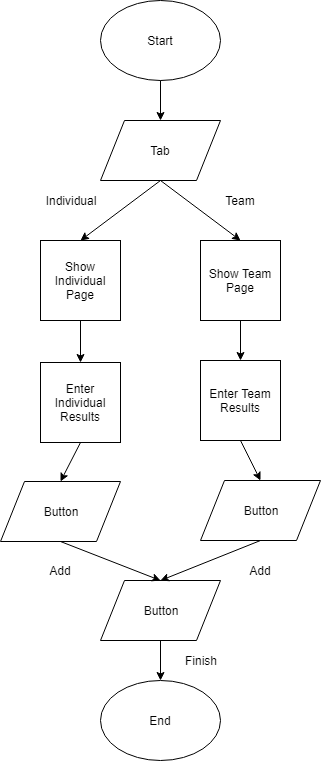
The points can be decided by us as they have not yet been decided.

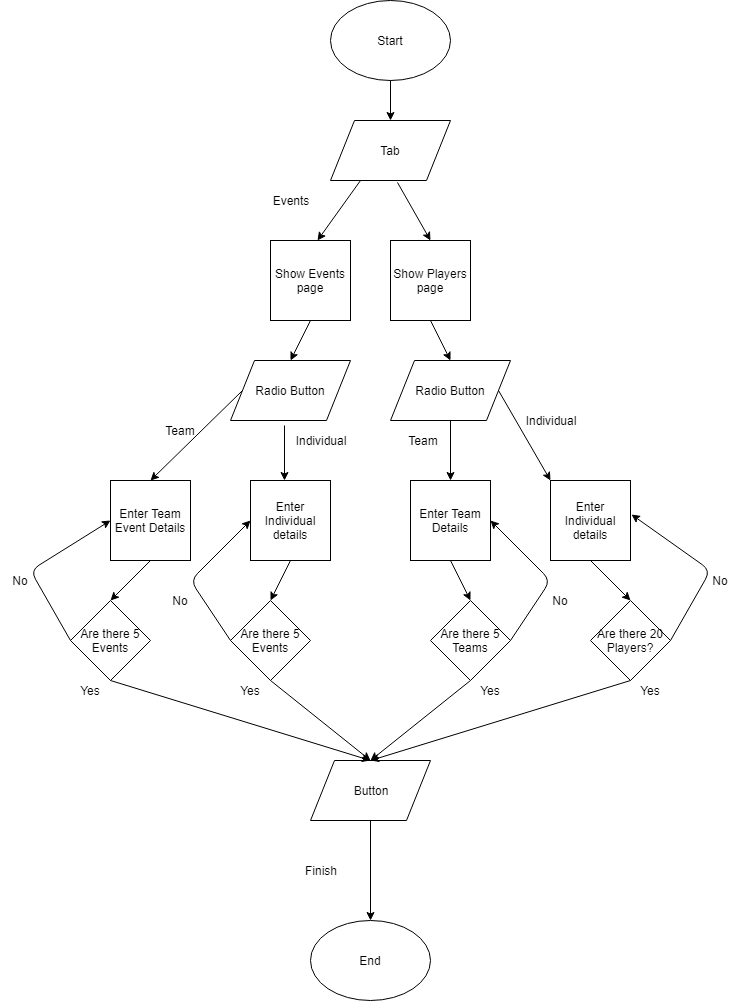
The college would like the ability for a single event to be entered on its own.

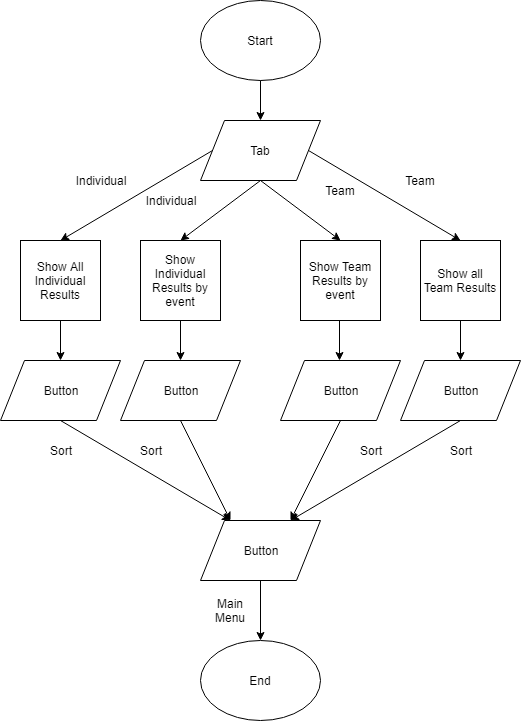
**Data Dictionary**

|  |  |  |
| --- | --- | --- |
| **Name** | **Form** | **Description** |
| btnStartNew | frmMainMenu | This will open the creation of the tournament. |
| btnLoad | frmMainMenu | This will load a previous tournament. |
| btnResultsAnalysis | frmMainMenu | This will show the Results. |
| btnEnterResults | frmMainMenu | This is to enter results. |
| lblNumInEv | frmSetup | This will update dependant on the number of individual events. |
| lblNumTeEv | frmSetup | This will update dependant on the number of team events. |
| lblIndSoFar | frmSetup | This shows the number of individuals so far. |
| lblTeamsSoFar | frmSetup | This shows the number of teams so far. |
| txtNames | frmSetup | Allows the user to write names in the box. |
| txtDetails | frmSetup | Allows the user to write the details in the box. |
| optTeam | frmSetup | Allows the ability to choose whether the event is team or individual |
| optIndividual | frmSetup | Allows the ability to choose whether the event is team or individual |
| btnAddDet | frmSetup | Allows the data to be added to the list box |
| lblNameIndv | frmSetup | Allows names to be written in the textbox |
| optTeamPar | frmSetup | Allows the option for Team to be selected |
| optIndPar | frmSetup | Allows the option for Individual to be selected |
| lstPartAdded | frmSetup | Allows the data to be added |
| lstEventsAdded | frmSetup | Allows the data to be added |
| lblTeamMembers | frmSetup | Allows the Team option to be selected and have data input. |
| txtTeamMem1 | frmSetup | Input box for Team member 1 |
| txtTeamMem2 | frmSetup | Input box for Team member 2 |
| txtTeamMem3 | frmSetup | Input box for Team member 3 |
| txtTeamMem4 | frmSetup | Input box for Team member 4 |
| txtTeamMem5 | frmSetup | Input box for Team member 5 |
| btnAddPar | frmSetup | Allows the data to be added to the list box |
| tbcRankings | frmRankings | This allows the user to choose the tab to view different information |
| txtInd1 | frmRankings | Input box for the independent player’s name |
| txtInd2 | frmRankings | Input box for the independent player’s name |
| txtInd3 | frmRankings | Input box for the independent player’s name |
| txtInd4 | frmRankings | Input box for the independent player’s name |
| txtInd5 | frmRankings | Input box for the independent player’s name |
| txtInd6 | frmRankings | Input box for the independent player’s name |
| txtInd7 | frmRankings | Input box for the independent player’s name |
| txtInd8 | frmRankings | Input box for the independent player’s name |
| txtInd9 | frmRankings | Input box for the independent player’s name |
| txtInd10 | frmRankings | Input box for the independent player’s name |
| txtInd11 | frmRankings | Input box for the independent player’s name |
| txtInd12 | frmRankings | Input box for the independent player’s name |
| txtInd13 | frmRankings | Input box for the independent player’s name |
| txtInd14 | frmRankings | Input box for the independent player’s name |
| txtInd15 | frmRankings | Input box for the independent player’s name |
| txtInd16 | frmRankings | Input box for the independent player’s name |
| txtInd17 | frmRankings | Input box for the independent player’s name |
| txtInd18 | frmRankings | Input box for the independent player’s name |
| txtInd19 | frmRankings | Input box for the independent player’s name |
| txtInd20 | frmRankings | Input box for the independent player’s name |
| btnSaveInd | frmRankings | To save all of the data |
| txtTeam1 | frmRankings | Input box for the Team’s name |
| txtTeam2 | frmRankings | Input box for the Team’s name |
| txtTeam3 | frmRankings | Input box for the Team’s name |
| txtTeam4 | frmRankings | Input box for the Team’s name |
| btnSaveTeam | frmRankings | Button to save the team |
| tbcAnalysis | frmAnalysis | This allows the user to choose the tab to view different information |
| cboIndEveRes | frmAnalysis | This combo box allows for the user to select an event |
| cboTeamEveRes | frmAnalysis | This combo box allows for the user to select an event |
| lstIndEveRes | frmAnalysis | Allows the data to be added |
| lstTeamEveRes | frmAnalysis | Allows the data to be added |
| btnSortIndEveRes | frmAnalysis | A button to sort the results |
| btnSortTeamEveRes | frmAnalysis | A button to sort the results |
| btnMainMenu1 | frmAnalysis | This button will return the user to the main menu |

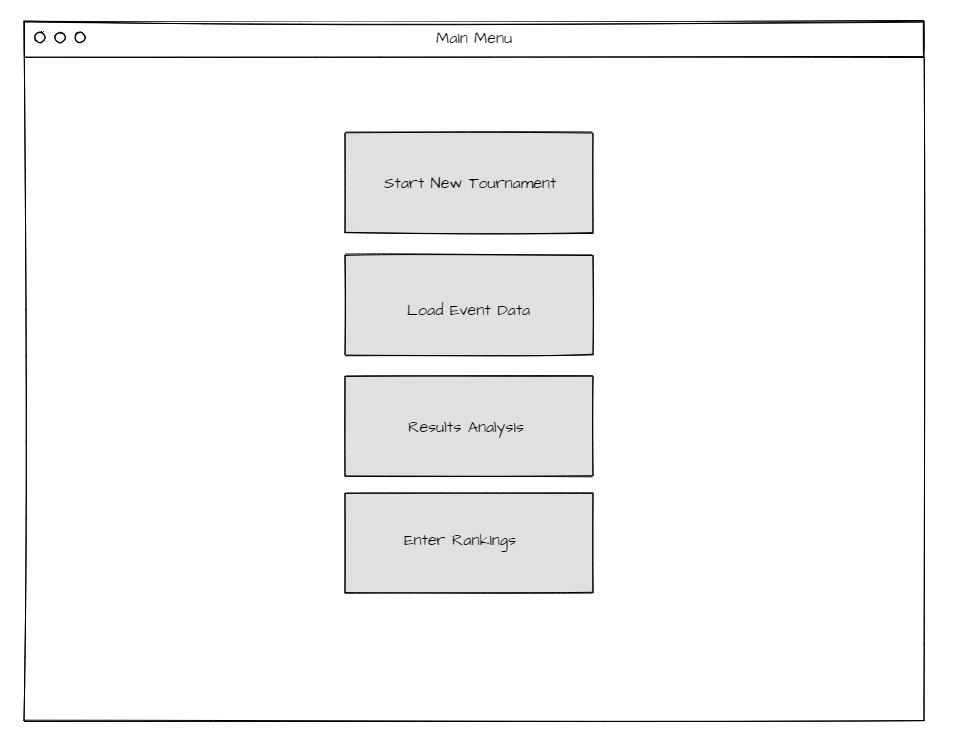
**Flow Charts:** 

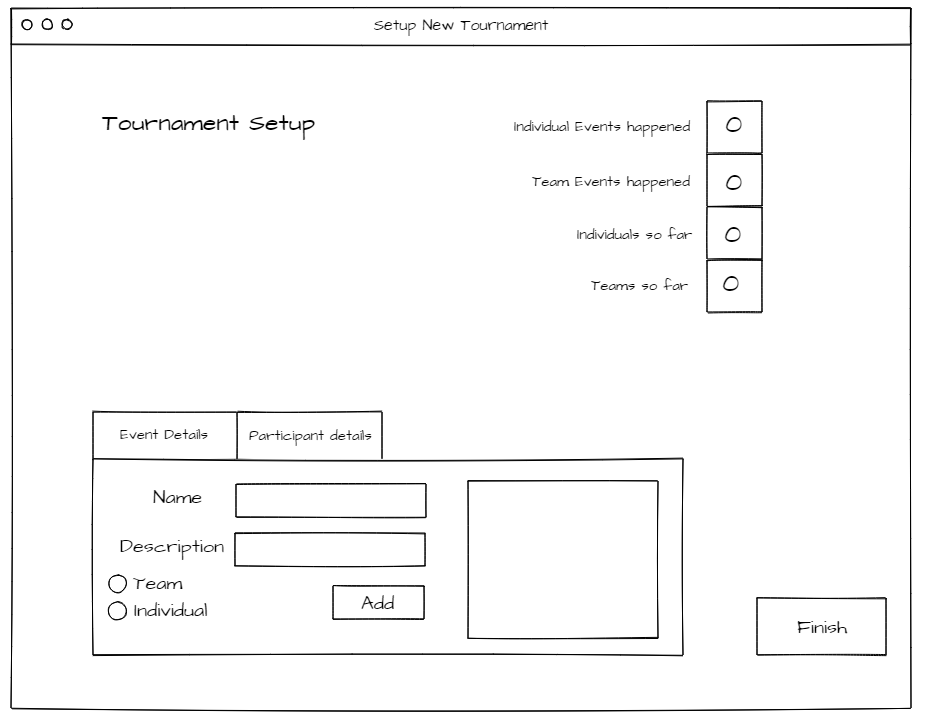


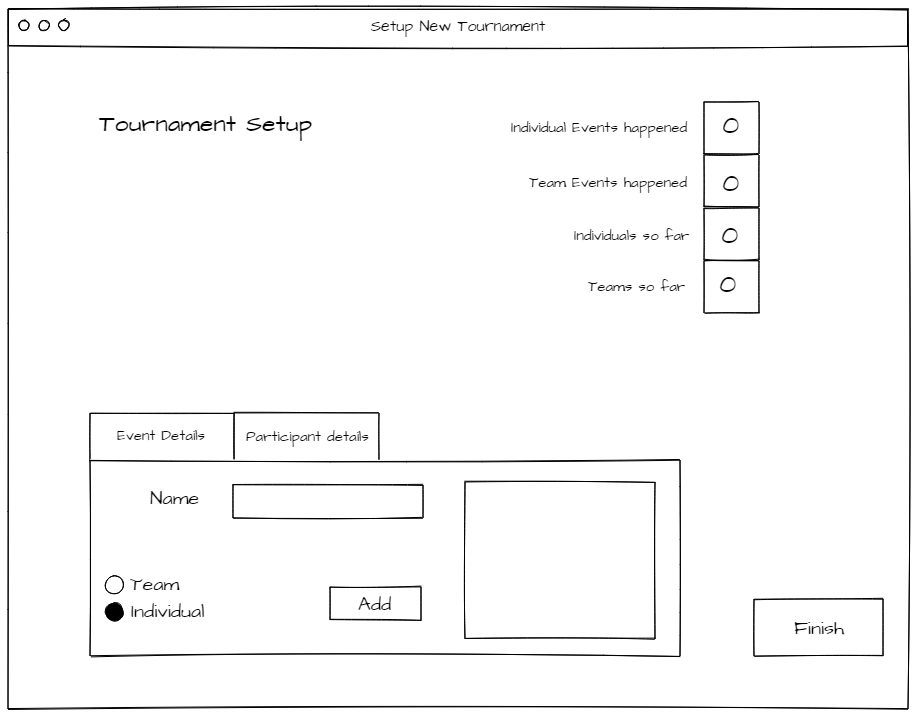




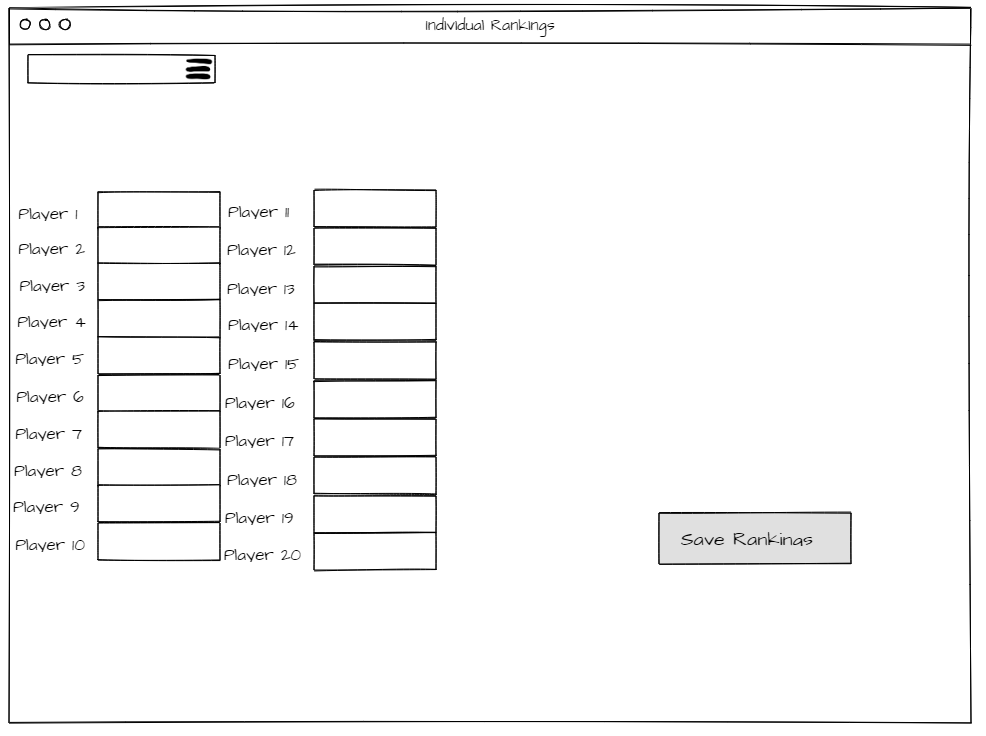
**Wireframes/Mock-ups:**

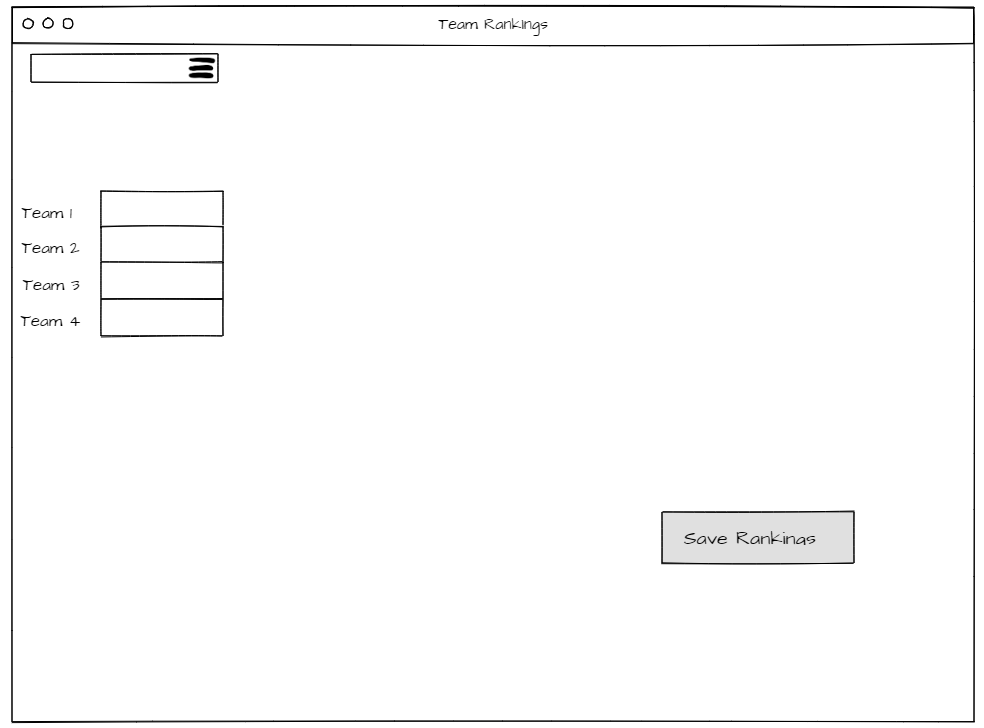
This is frmHome:

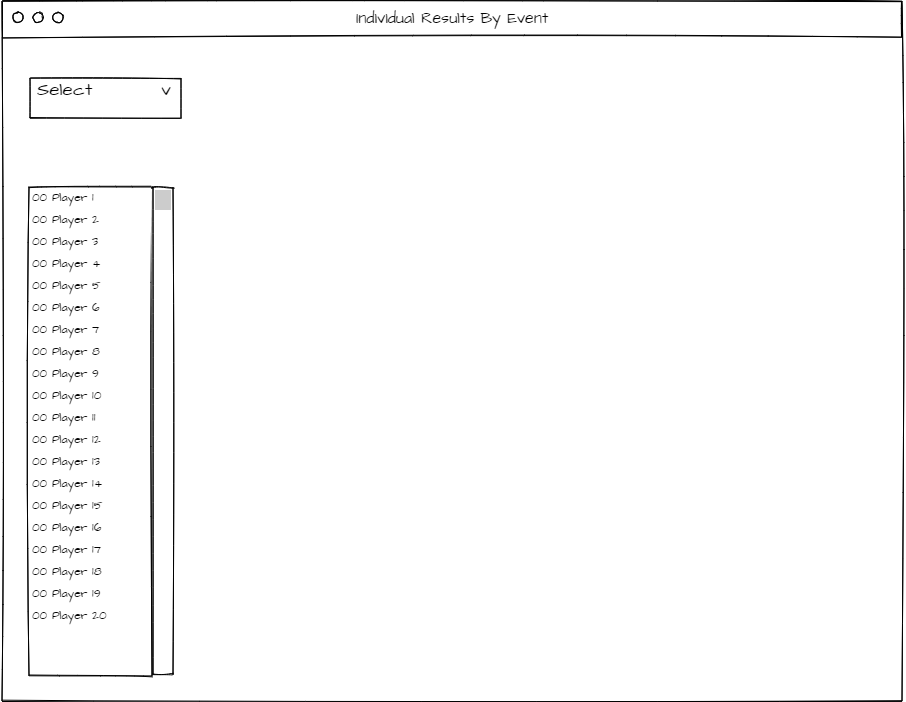
This is frmSetup:

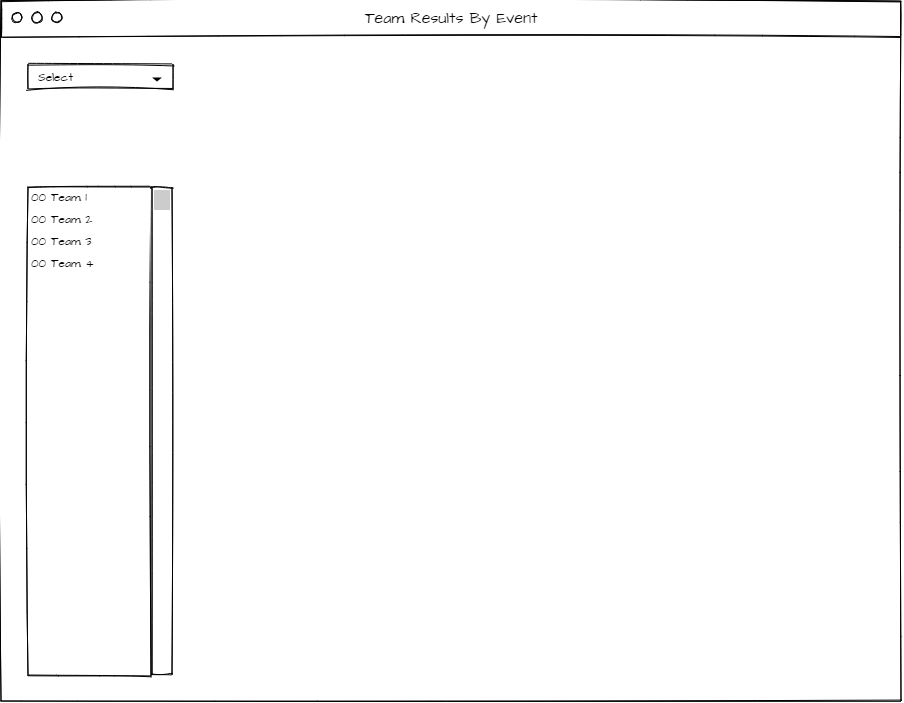
This is frmSetup: 

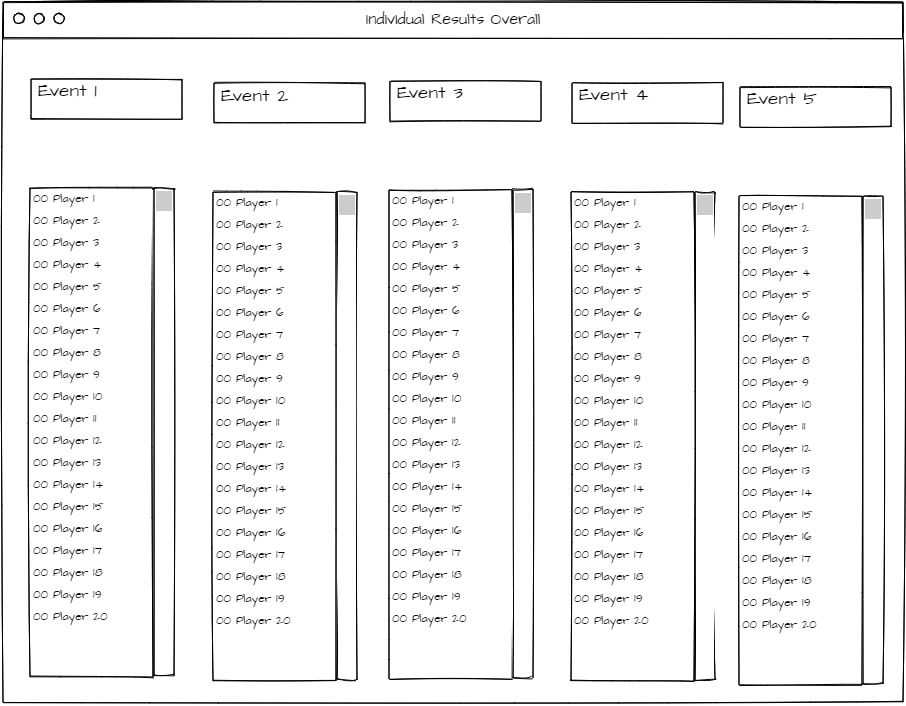
This is frmSetup: 

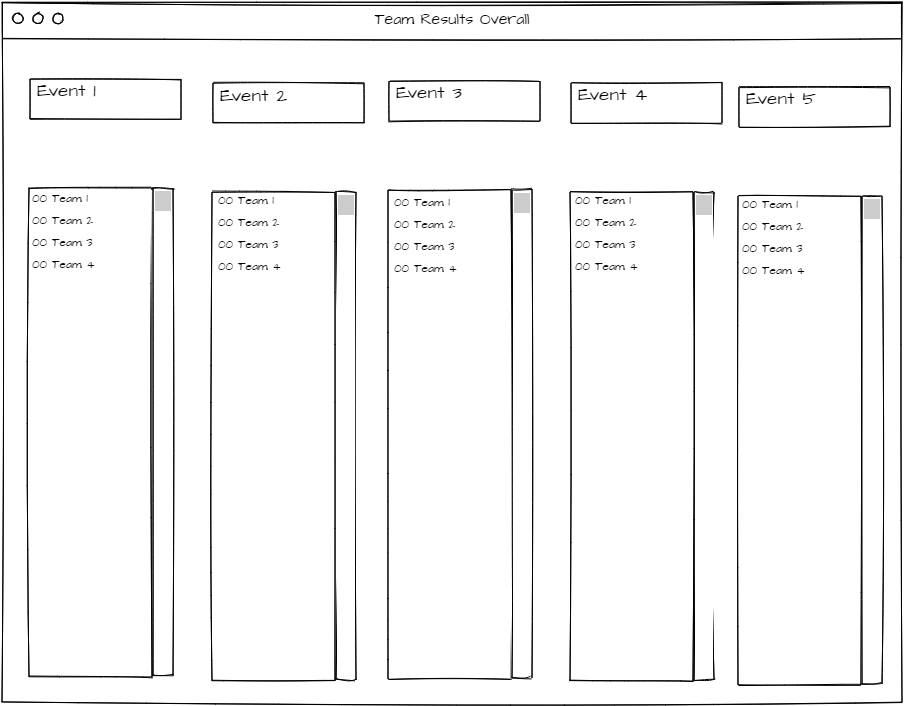
This is frmRankings: 

This is frmRankings: 

This is frmAnalysis: 

This is frmAnalysis: 

This is frmAnalysis: 

This is frmAnalysis: 

**Hardware**

For this application, the users need to have a very minimum of hardware. They do not need complicatedly powerful CPUs or GPUs:

They do not need a high amount or RAM.

They will need a mouse.

They will need a keyboard.

**Software**

In terms of software, the application will be an executable program and therefore they will not need the Visual Studio to use it.

The user will need an acceptable version of Windows such as Windows 7, Windows 8 or Windows 10.

**Review**

**User Questionnaire**

1) Is the text readable and easy to interpret?

A) Yes, your text is readable. You may want to use a sans-serif font for it to be easily readable by the user.

A2) Yes it is.

2) Does the program include all of the needed information?

A) The program has all of the required needed information in order to function.

A2) Yes, the program has all of the information it needs.

3) Does the program look professional?

A) The programs could be made professional with the use of a colour scheme or the college logo within the program.

A2) The programs follow a consistent scheme and therefore looks professional.

4) Are all of the designs consistent, preventing any confusion?

A) The designs are consistent although only some of the pages have buttons so the user will not be able to navigate from the sections. This is visible on the result pages.

A2)

5) Are different forms identifiable (do they have the correct names etc.)

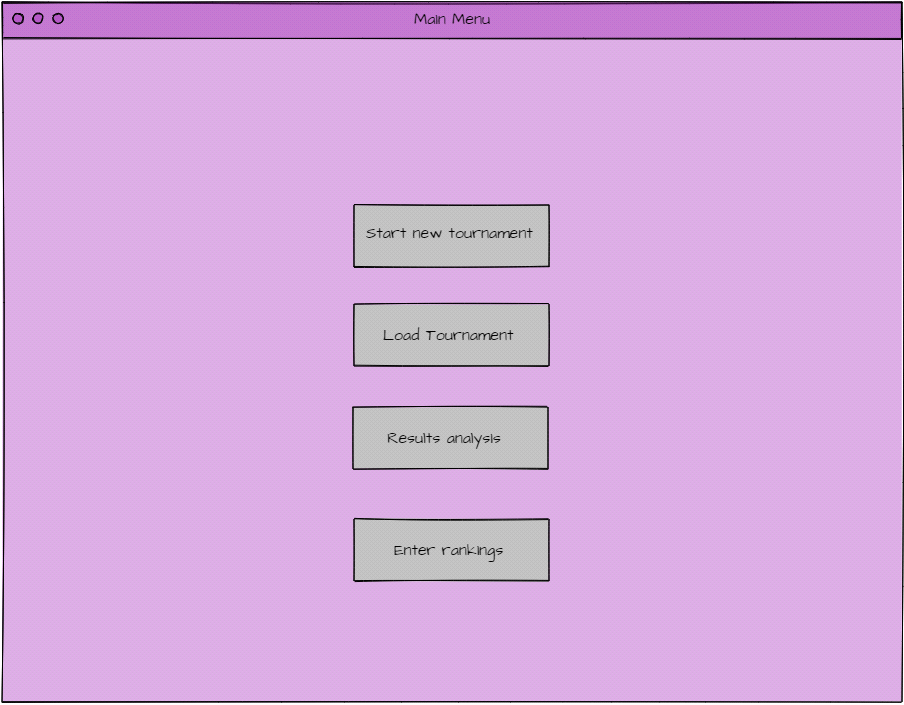
A) Yes, the forms have different names relating to the context of the form.

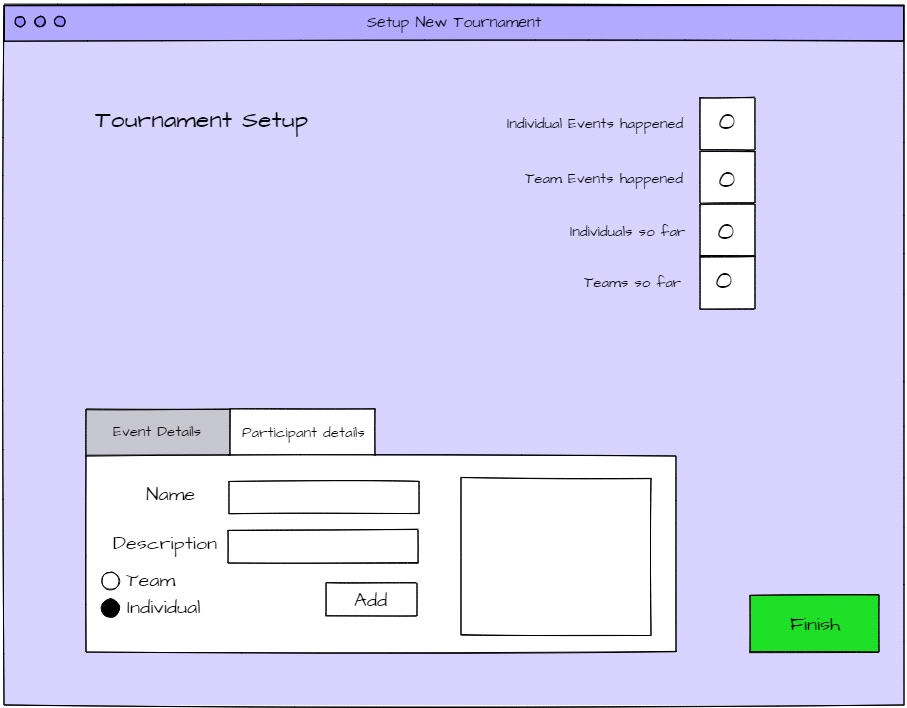
A2) The forms a relative to the titles.

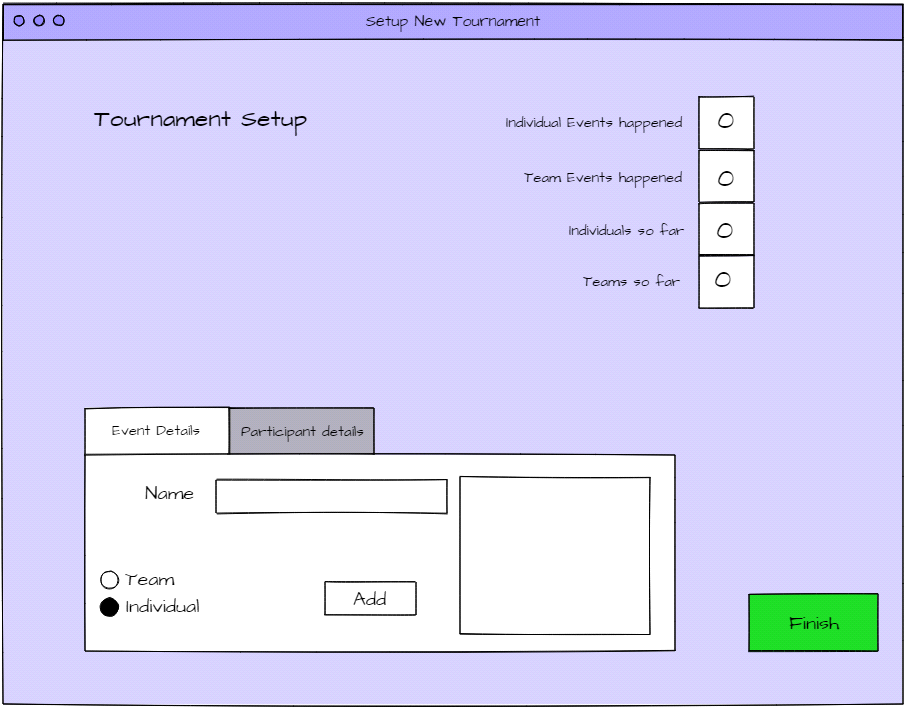
6) Is the information in necessary areas so that they do not cause any confusion?

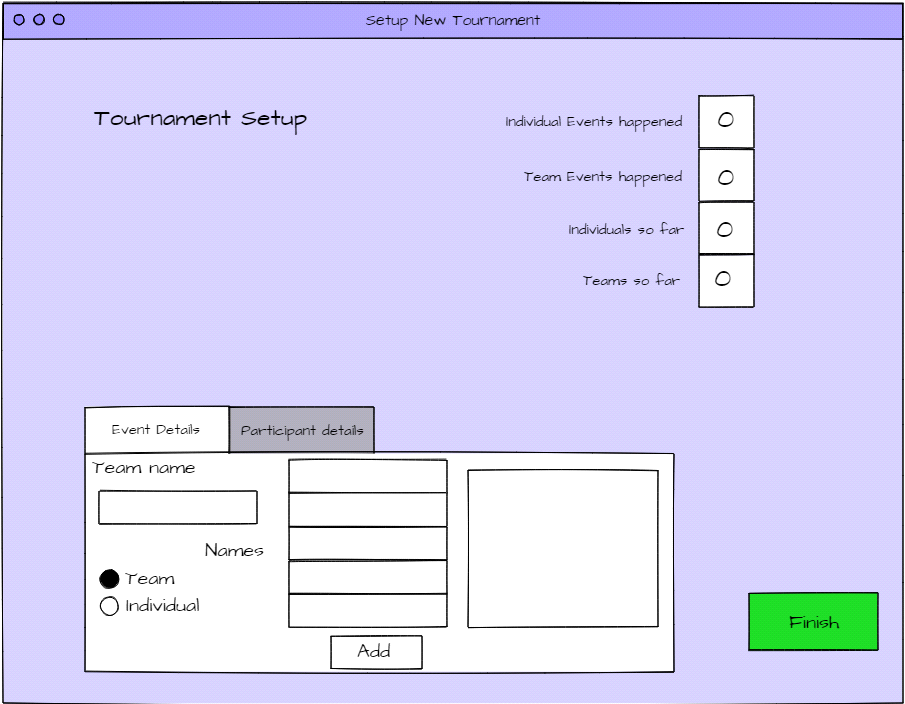
**Improvements to Design**

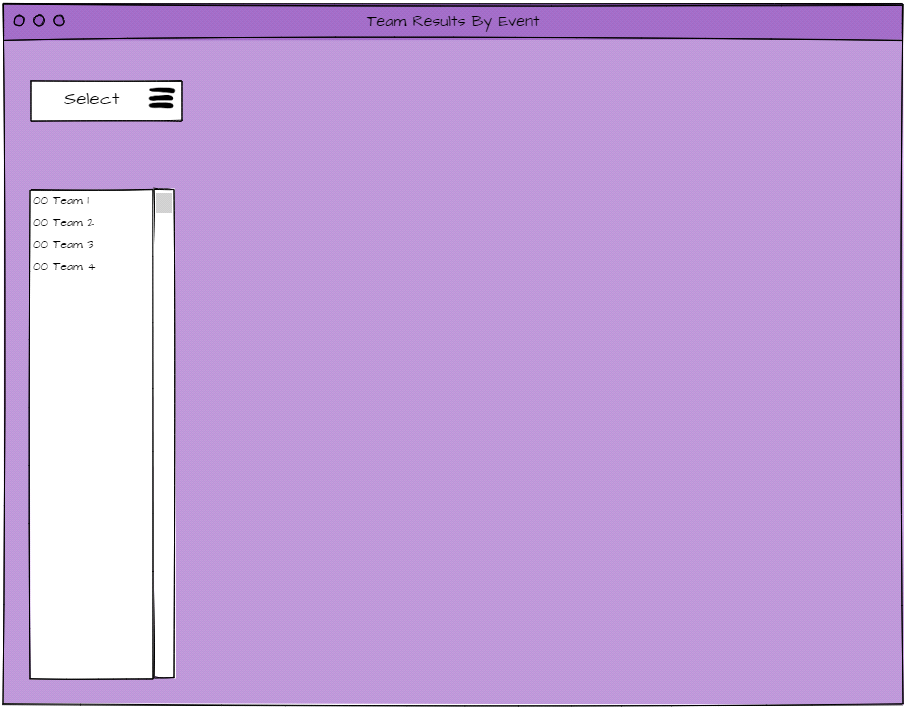
Menu disabling buttons

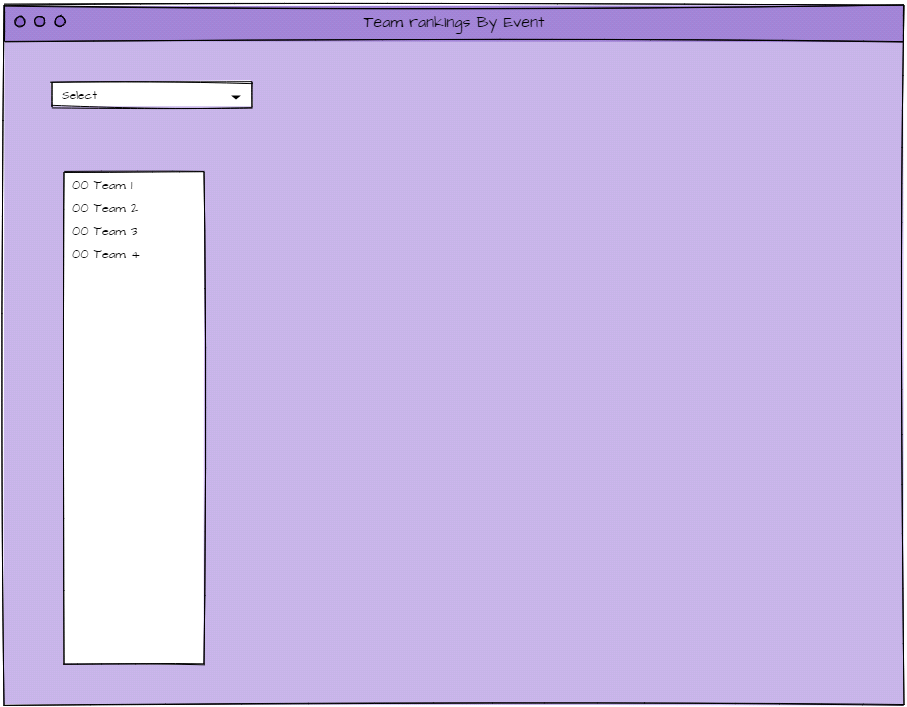
frmHome: 

frmSetup: 

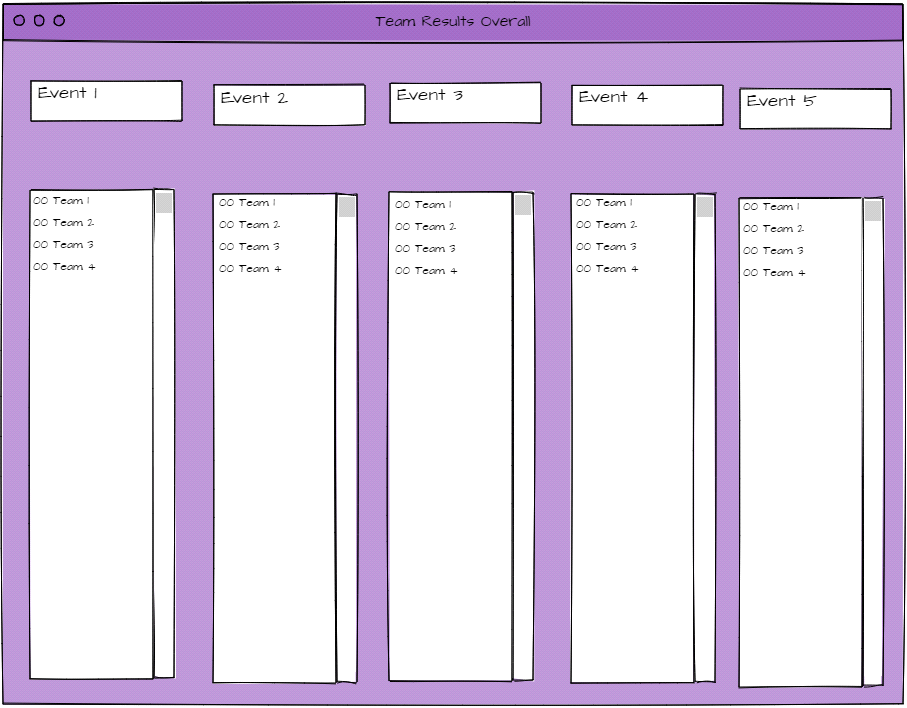
frmSetup: 

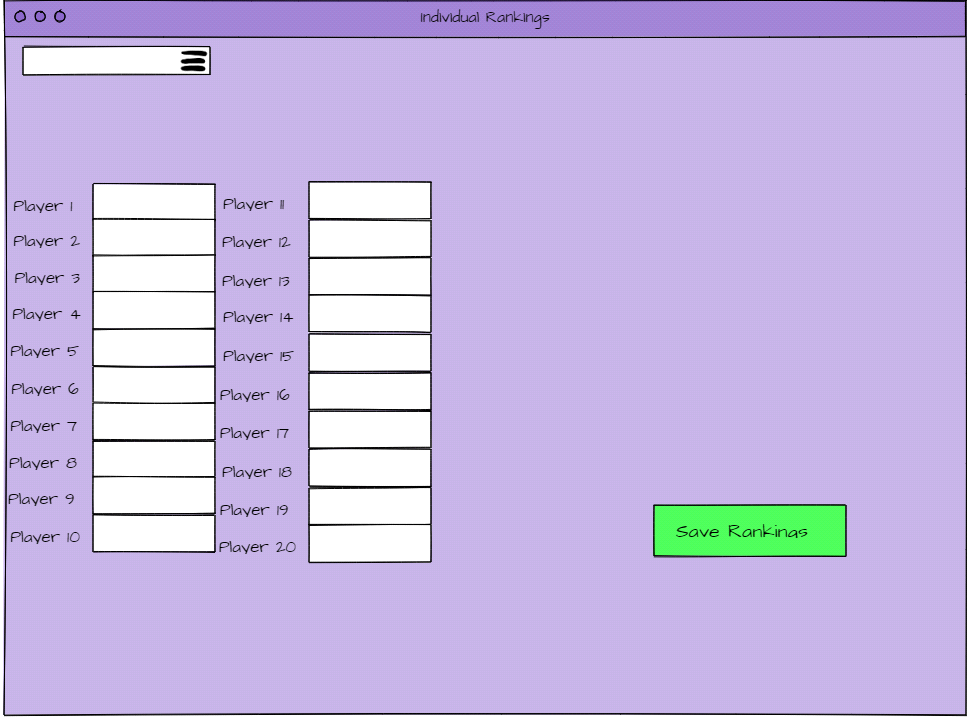
frmSetup: 

frmAnalysis: 

frmAnalysis: 

frmAnalysis: 

frmAnalysis: 

frmRankings: 

frmRankings: 

**Development**

**Forms and Code:**

Module Globals

Structure IndividualPlayerType

Dim iName As String

Dim iEvent() As String

Dim iResult() As Integer

Dim iOneEventer As Boolean

End Structure

Structure TeamPlayerType

Dim TName As String

Dim TPlayerName() As String

Dim TEvent() As String

Dim TResult() As Integer

Dim TOneEventer As Boolean

End Structure

Structure TournamentEventsType

Dim EName As String

Dim EDesc As String

End Structure

Public ind(19) As IndividualPlayerType

Public team(3) As TeamPlayerType

Public TEvent(5) As TournamentEventsType

Public IEvent(5) As TournamentEventsType

Public TEventCount As Byte = 0

Public IEventCount As Byte = 0

Public indCount As Byte = 0

Public teamCount As Byte = 0

End Module

**Form1:**

Public Class Form1

Private Sub btnStartNew\_Click(sender As Object, e As EventArgs) Handles btnStart.Click

Me.Hide()

frmSetup.Show()

End Sub

Private Sub btnLoad\_Click(sender As Object, e As EventArgs) Handles btnLoad.Click

'When this button is pressed, it will load preset data

Dim i As Byte

'Team events

TEvent(0).EName = "Football"

TEvent(1).EName = "Basketball"

TEvent(2).EName = "Rugby"

TEvent(3).EName = "Baseball"

TEvent(4).EName = "Cricket"

'Individual events

IEvent(0).EName = "Table Tennis"

IEvent(1).EName = "Tennis"

IEvent(2).EName = "Sprint"

IEvent(3).EName = "Golf"

IEvent(4).EName = "Javelin"

'Changes the counters to reflect the amount of events

TEventCount = 5

IEventCount = 5

indCount = 20

teamCount = 4

'Adding individuals to the tournament setting their scores to zero

For i = 0 To 19

ind(i).iName = "Player" & i + 1

ind(i).iEvent = {IEvent(0).EName, IEvent(1).EName, IEvent(2).EName, IEvent(3).EName, IEvent(4).EName}

ind(i).iResult = {0, 0, 0, 0, 0}

Next i

'Adding teams to the tournament setting their scores to zero

For i = 0 To 3

team(i).TName = "Team " & Chr(65 + i)

team(i).TPlayerName = {"Team " & Chr(65 + i) & " - Player1",

"Team " & Chr(65 + i) & " - Player2",

"Team " & Chr(65 + i) & " - Player3",

"Team " & Chr(65 + i) & " - Player4",

"Team " & Chr(65 + i) & " - Player5"}

team(i).TEvent = {TEvent(0).EName, TEvent(1).EName, TEvent(2).EName, TEvent(3).EName, TEvent(4).EName}

team(i).TResult = {0, 0, 0, 0, 0}

Next i

End Sub

Private Sub btnResults\_Click(sender As Object, e As EventArgs) Handles btnResults.Click

Me.Hide()

FrmAnalysis.Show()

End Sub

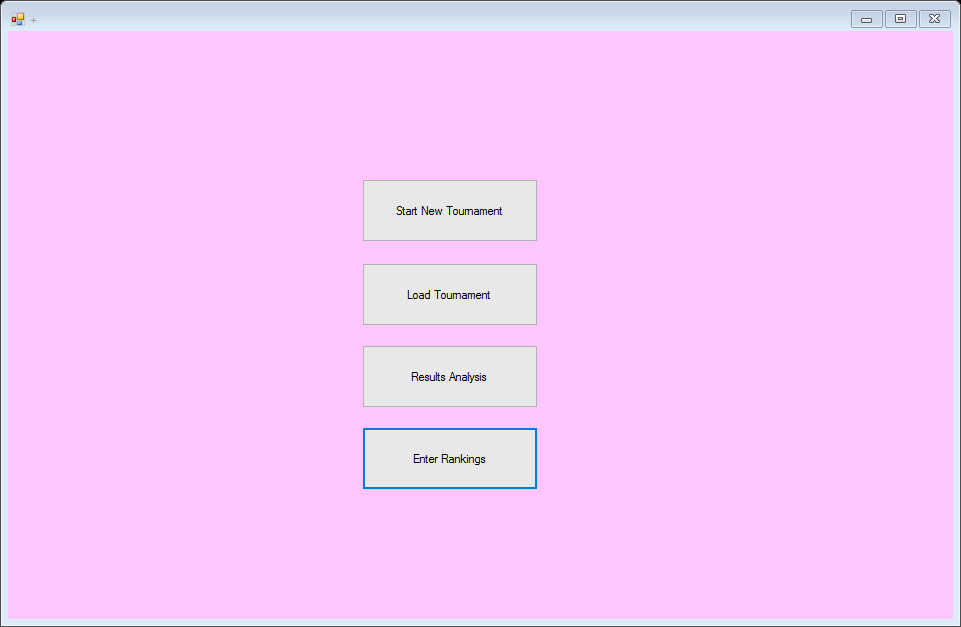
Private Sub btnEnterRankings\_Click(sender As Object, e As EventArgs) Handles btnEnterRankings.Click

Me.Hide()

frmRankings.Show()

End Sub

End Class



**frmSetup:**

Public Class frmSetup

Private Sub frmSetup\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'Displays count of events as label on form

lblNumInEv.Text = IEventCount

lblNumTeEv.Text = TEventCount

lblIndSoFar.Text = indCount

lblTeamsSoFar.Text = teamCount

'Clears text boxes

txtEventName.Text = ""

txtEventDesc.Text = ""

txtPartName.Text = ""

'Clears list box

lstEventsAdded.Items.Clear()

lstPartAdded.Items.Clear()

'Check if any events have been added

If IEventCount <> 0 Then

'Adds events to listbox

For i = 0 To IEventCount - 1

lstEventsAdded.Items.Add(IEvent(i).EName & " Ind")

Next

End If

If TEventCount <> 0 Then

For i = 0 To TEventCount - 1

lstEventsAdded.Items.Add(TEvent(i).EName & " Team")

Next

End If

If indCount <> 0 Then

'Adds participants to listbox

For i = 0 To indCount - 1

lstPartAdded.Items.Add(ind(i).iName & " Ind")

Next

End If

If teamCount <> 0 Then

For i = 0 To teamCount - 1

lstPartAdded.Items.Add(team(i).TName & " Team")

Next

End If

End Sub

Private Sub btnAddDet\_Click(sender As Object, e As EventArgs) Handles btnAddDet.Click

'Checks for the option individual selected

If optIndividual.Checked = True Then

'Checks if there is enough room in the listbox for another event

If IEventCount < 5 Then

IEvent(IEventCount).EName = txtEventName.Text

'input for event name

IEvent(IEventCount).EDesc = txtEventDesc.Text

'input for description

IEventCount += 1

'adds 1

lblNumInEv.Text = IEventCount

'adds 1

lstEventsAdded.Items.Add(txtEventName.Text & " Ind")

'Adds the item to the listbox

Else

'Makes the user aware there are too many events added

MsgBox("There are already 5 individual events.")

End If

End If

'Checks for the option team selected

If optTeam.Checked = True Then

'Checks if there is enough room in the listbox for another event

If TEventCount < 5 Then

TEvent(TEventCount).EName = txtEventName.Text

'input for event name

TEvent(TEventCount).EDesc = txtEventDesc.Text

'input for description

TEventCount += 1

'adds 1

lblNumTeEv.Text = TEventCount

'Adds the item to the listbox

lstEventsAdded.Items.Add(txtEventName.Text & " Team")

'Makes the user aware there are too many events added

Else

MsgBox("There are already 5 Team events.")

End If

End If

'Resets text boxes

txtEventName.Text = ""

txtEventDesc.Text = ""

End Sub

Private Sub rdoIndP\_CheckedChanged(sender As Object, e As EventArgs) Handles optIndPar.CheckedChanged

'If individual participant selected, hide the following

If optIndPar.Checked = True Then

txtTeamMem1.Visible = False

txtTeamMem2.Visible = False

txtTeamMem3.Visible = False

txtTeamMem4.Visible = False

txtTeamMem5.Visible = False

lblTeamMembers.Visible = False

'If Team selected, show the following

Else

txtTeamMem1.Visible = True

txtTeamMem2.Visible = True

txtTeamMem3.Visible = True

txtTeamMem4.Visible = True

txtTeamMem5.Visible = True

lblTeamMembers.Visible = True

End If

End Sub

Private Sub btnAddPar\_Click(sender As Object, e As EventArgs) Handles btnAddPar.Click

'Checks that the details are correctly completed

If txtPartName.Text <> "" Then

If optIndPar.Checked = True Then

If indCount < 20 Then

ind(indCount).iName = txtPartName.Text

ind(indCount).iResult = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

indCount += 1

lblIndSoFar.Text = indCount

lstPartAdded.Items.Add(txtPartName.Text & " Ind")

Else

MsgBox("Max number of individual players has been reached.")

End If

End If

'checks the radio button for Team selected

If optTeamPar.Checked = True Then

'Asks for 5 names

If txtTeamMem1.Text = "" Or txtTeamMem2.Text = "" Or txtTeamMem3.Text = "" Or txtTeamMem4.Text = "" Or txtTeamMem5.Text = "" Then

MsgBox("Please enter 5 names for the team")

Else

'Checks that there are less than 4 teams

If teamCount < 4 Then

'Team name

team(teamCount).TName = txtPartName.Text

'Team members

team(teamCount).TPlayerName = {txtTeamMem1.Text, txtTeamMem2.Text, txtTeamMem3.Text, txtTeamMem4.Text, txtTeamMem5.Text}

'Increases the team count since a team has been added.

team(teamCount).TResult = {0, 0, 0, 0, 0}

teamCount += 1

lblTeamsSoFar.Text = teamCount

'Adds the team to the listbox

lstPartAdded.Items.Add(txtPartName.Text & " Team")

Else

'Personalised message for when max num teams been added

MsgBox("Max number of teams has been reached.")

End If

End If

End If

Else

MsgBox("Please enter the name of the individual")

End If

'Clears textbox for teamn/player name

txtPartName.Text = ""

'Clear textboxes for player names

txtTeamMem1.Text = ""

txtTeamMem2.Text = ""

txtTeamMem3.Text = ""

txtTeamMem4.Text = ""

txtTeamMem5.Text = ""

End Sub

Private Sub BtnFinishSetup\_Click(sender As Object, e As EventArgs) Handles BtnFinishSetup.Click

'Checks if correct number of events and participants are present

If TEventCount <> 5 Or IEventCount <> 5 Or indCount <> 20 Or teamCount <> 4 Then

MsgBox("Please check that you have entered 5 individual events, 5 team events, 20 individual participants and 4 teams just before clicking Finish!")

Else

'Disables and enables correct buttons

'Closes form and navigates back to Main Menu

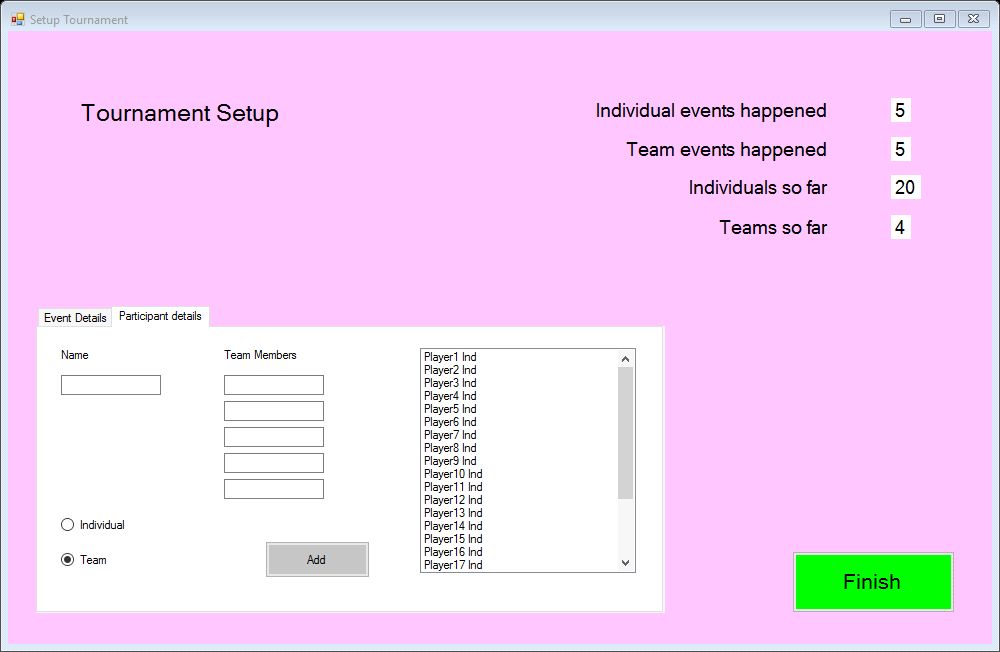
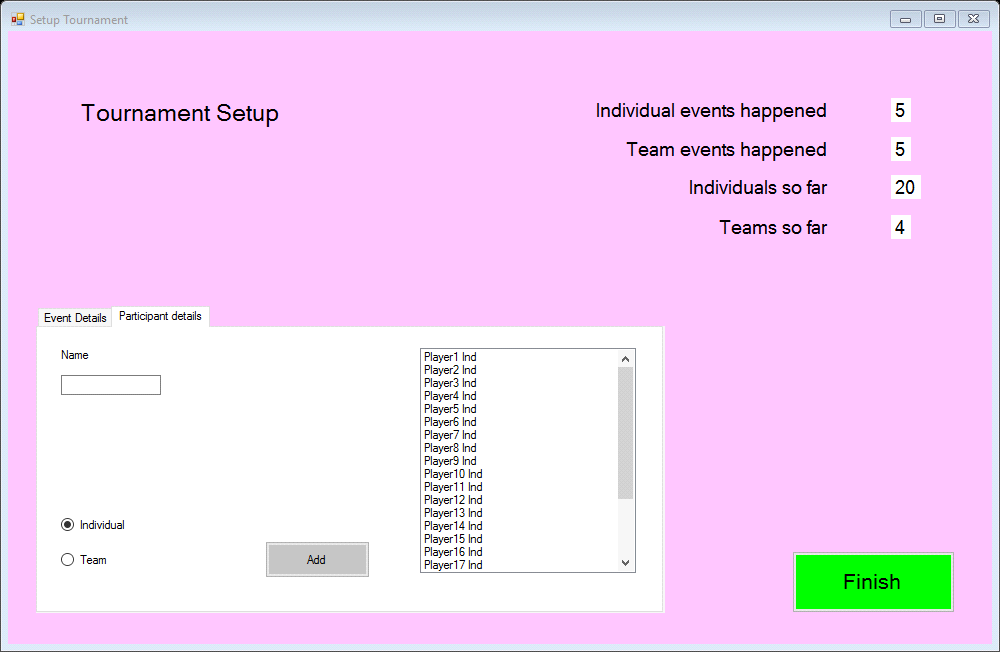
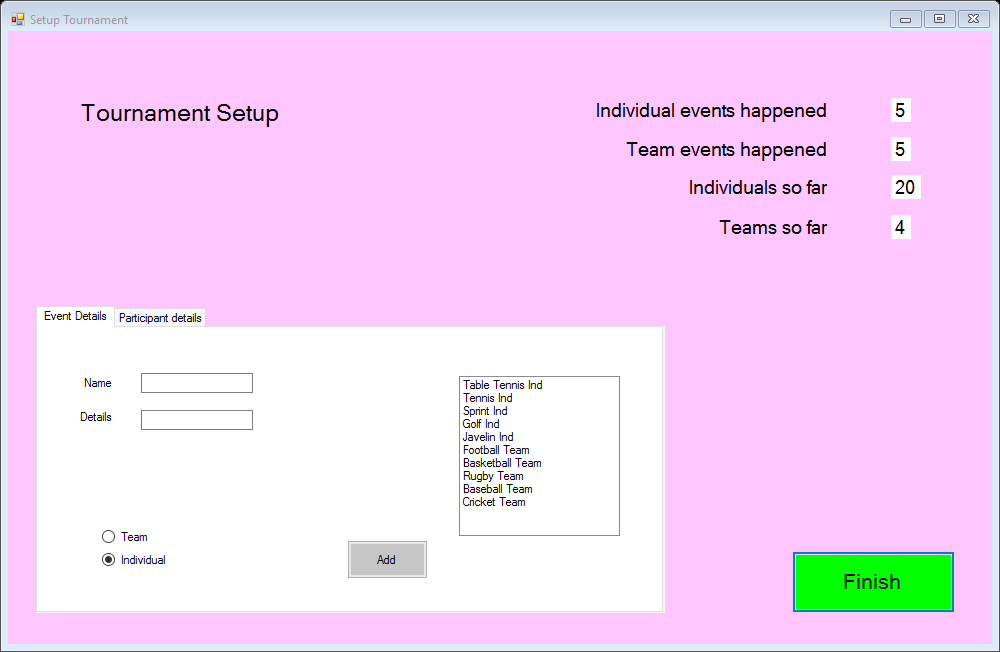
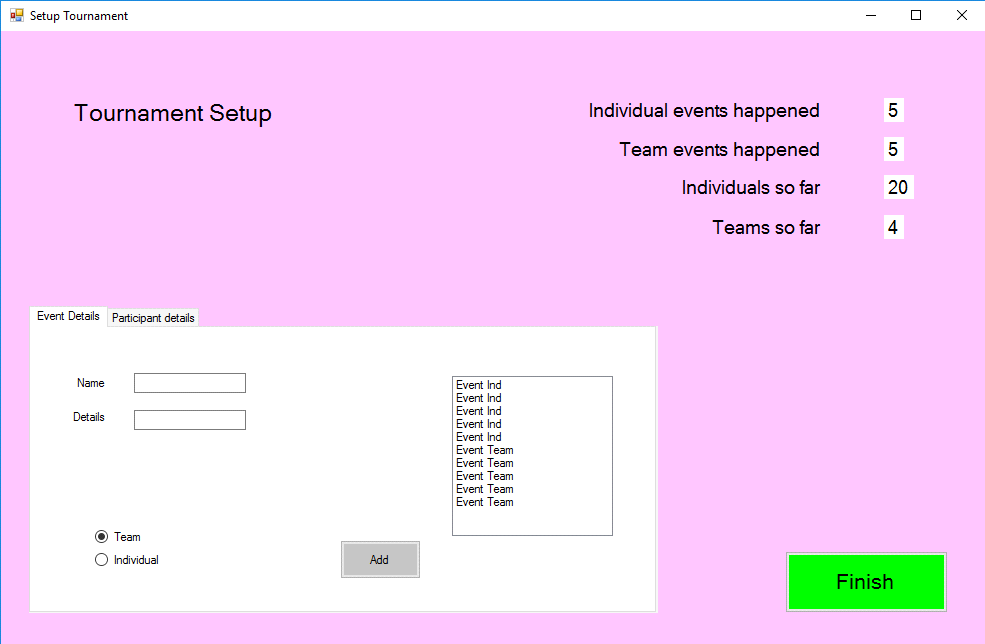
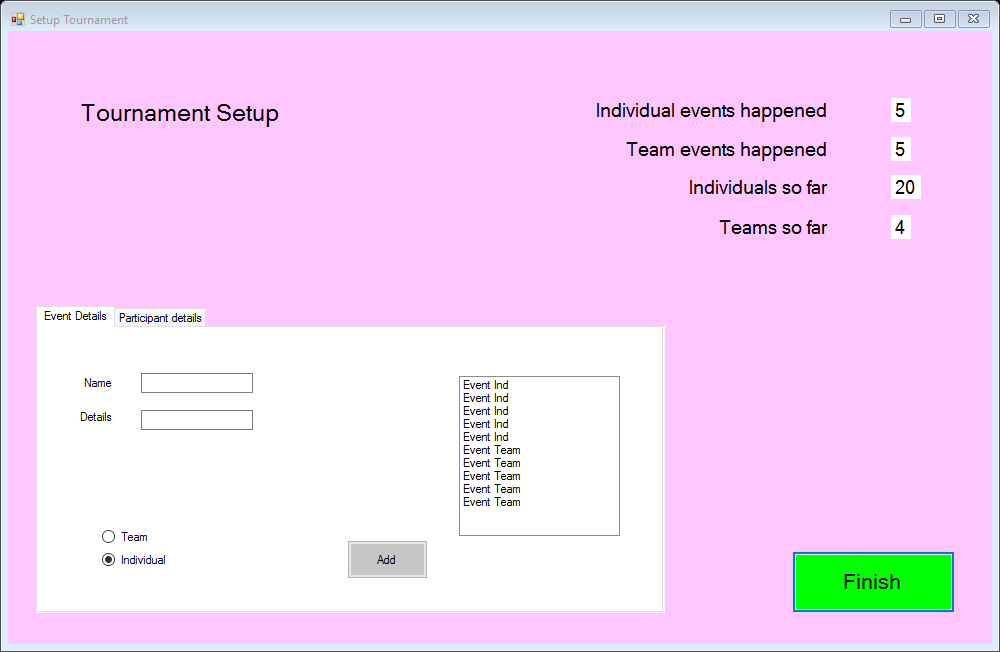
Me.Close()

Form1.Show()

End If

End Sub

End Class



**frmAnalysis:** Public Class FrmAnalysis

Private Sub frmAnalysis\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'Setup combo boxes

Dim i As Byte

' Adds Individual events to Combo box

For i = 0 To 4

cboIndEveResults.Items.Add(IEvent(i).EName)

Next i

'Setup combo boxes

Dim j As Byte

' Adds team events to Combo box

For j = 0 To 4

cboTeamEveRes.Items.Add(TEvent(j).EName)

Next j

'Clears list if there are any old results

lstIndRes.Items.Clear()

For x = 0 To 19

'Setup listboxes with results

Dim tempiresult As Integer

tempiresult = ind(x).iResult(0) + ind(x).iResult(1) + ind(x).iResult(2) + ind(x).iResult(3) + ind(x).iResult(4)

lstIndRes.Items.Add(Format(tempiresult, "00") & " " & ind(x).iName)

Next x

'Clears list if there are any old results

lstTeamRes.Items.Clear()

For y = 0 To 3

'Setup listbox with adding team results

Dim temptresult As Integer

temptresult = team(y).TResult(0) + team(y).TResult(1) + team(y).TResult(2) + team(y).TResult(3) + team(y).TResult(4)

lstTeamRes.Items.Add(Format(temptresult, "00") & " " & team(y).TName)

Next y

End Sub

Private Sub ComboBox1\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cboIndEveResults.SelectedIndexChanged

lstIndEveResults.Items.Clear()

'Selects correct case to add results to listbox

Select Case cboIndEveResults.SelectedIndex

Case 0

For i = 0 To 19

lstIndEveResults.Items.Add(Format(ind(i).iResult(0), "00") & " " & ind(i).iName)

Next i

Case 1

For i = 0 To 19

lstIndEveResults.Items.Add(Format(ind(i).iResult(1), "00") & " " & ind(i).iName)

Next i

Case 2

For i = 0 To 19

lstIndEveResults.Items.Add(Format(ind(i).iResult(2), "00") & " " & ind(i).iName)

Next i

Case 3

For i = 0 To 19

lstIndEveResults.Items.Add(Format(ind(i).iResult(3), "00") & " " & ind(i).iName)

Next i

Case 4

For i = 0 To 19

lstIndEveResults.Items.Add(Format(ind(i).iResult(4), "00") & " " & ind(i).iName)

Next i

End Select

End Sub

Private Sub ComboBox2\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cboTeamEveRes.SelectedIndexChanged

lstTeamEveRes.Items.Clear()

'Selects correct case to add results to listbox

Select Case cboTeamEveRes.SelectedIndex

Case 0

For i = 0 To 3

lstTeamEveRes.Items.Add(Format(team(i).TResult(0), "00") & " " & team(i).TName)

Next i

Case 1

For i = 0 To 3

lstTeamEveRes.Items.Add(Format(team(i).TResult(1), "00") & " " & team(i).TName)

Next i

Case 2

For i = 0 To 3

lstTeamEveRes.Items.Add(Format(team(i).TResult(2), "00") & " " & team(i).TName)

Next i

Case 3

For i = 0 To 3

lstTeamEveRes.Items.Add(Format(team(i).TResult(3), "00") & " " & team(i).TName)

Next i

Case 4

For i = 0 To 3

lstTeamEveRes.Items.Add(Format(team(i).TResult(4), "00") & " " & team(i).TName)

Next i

End Select

End Sub

Private Sub btnSortIndEveRes\_Click(sender As Object, e As EventArgs) Handles btnSortIndEveRes.Click

Dim xResults() As String = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Dim xTemp As String

Dim i, swaps As Integer

'Sorts array of results by comparing what it has already sorted

For i = 0 To lstIndEveResults.Items.Count - 1

xResults(i) = lstIndEveResults.Items.Item(i)

Next

swaps = 1

While swaps > 0

'Swaps is set to zero, it is the counter for this loop. When the results are swapped, 1 is added meaning the while loop is > 0 and then loops again, until nothing was swapped and the while is not > 0

swaps = 0

For i = 0 To xResults.Count - 2

If xResults(i) < xResults(i + 1) Then

xTemp = xResults(i)

xResults(i) = xResults(i + 1)

xResults(i + 1) = xTemp

swaps += 1

End If

Next

End While

lstIndEveResults.Items.Clear()

For i = 0 To xResults.Count - 1

lstIndEveResults.Items.Add(xResults(i))

Next

End Sub

Private Sub btnSortTeamEveRes\_Click(sender As Object, e As EventArgs) Handles btnSortTeamEveRes.Click

Dim xResults() As String = {0, 0, 0, 0}

Dim xTemp As String

Dim i, swaps As Integer

'Sorts array of results by comparing what it has already sorted

For i = 0 To lstTeamEveRes.Items.Count - 1

xResults(i) = lstTeamEveRes.Items.Item(i)

Next

swaps = 1

While swaps > 0

'Swaps is set to zero, it is the counter for this loop. When the results are swapped, 1 is added meaning the while loop is > 0 and then loops again, until nothing was swapped and the while is not > 0

swaps = 0

For i = 0 To xResults.Count - 2

If xResults(i) < xResults(i + 1) Then

xTemp = xResults(i)

xResults(i) = xResults(i + 1)

xResults(i + 1) = xTemp

swaps += 1

End If

Next

End While

lstTeamEveRes.Items.Clear()

For i = 0 To xResults.Count - 1

lstTeamEveRes.Items.Add(xResults(i))

Next

End Sub

Private Sub btnSortIndRes\_Click(sender As Object, e As EventArgs) Handles btnSortIndRes.Click

Dim xResults() As String = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

Dim xTemp As String

Dim i, swaps As Integer

For i = 0 To lstIndRes.Items.Count - 1

xResults(i) = lstIndRes.Items.Item(i)

Next

swaps = 1

While swaps > 0

swaps = 0

For i = 0 To xResults.Count - 2

If xResults(i) < xResults(i + 1) Then

xTemp = xResults(i)

xResults(i) = xResults(i + 1)

xResults(i + 1) = xTemp

swaps += 1

End If

Next

End While

lstIndRes.Items.Clear()

For i = 0 To xResults.Count - 1

lstIndRes.Items.Add(xResults(i))

Next

End Sub

Private Sub btnSortTeamRes\_Click(sender As Object, e As EventArgs) Handles btnSortTeamRes.Click

Dim xResults() As String = {0, 0, 0, 0}

Dim xTemp As String

Dim i, swaps As Integer

For i = 0 To lstTeamRes.Items.Count - 1

xResults(i) = lstTeamRes.Items.Item(i)

Next

swaps = 1

While swaps > 0

swaps = 0

For i = 0 To xResults.Count - 2

If xResults(i) < xResults(i + 1) Then

xTemp = xResults(i)

xResults(i) = xResults(i + 1)

xResults(i + 1) = xTemp

swaps += 1

End If

Next

End While

lstTeamRes.Items.Clear()

For i = 0 To xResults.Count - 1

lstTeamRes.Items.Add(xResults(i))

Next

End Sub

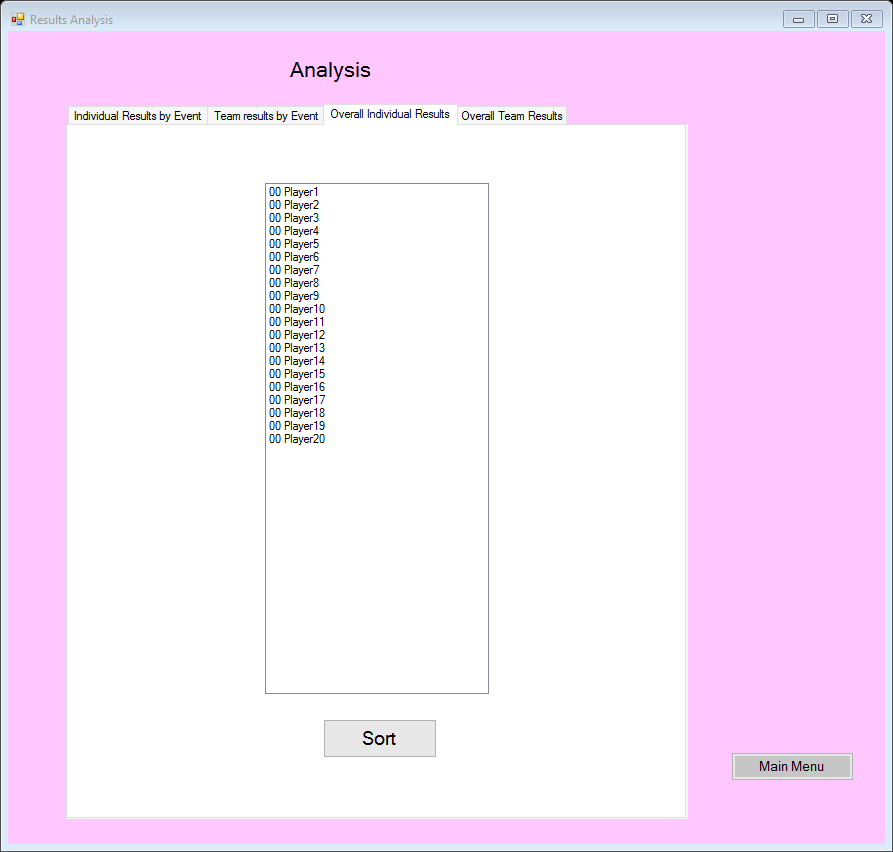
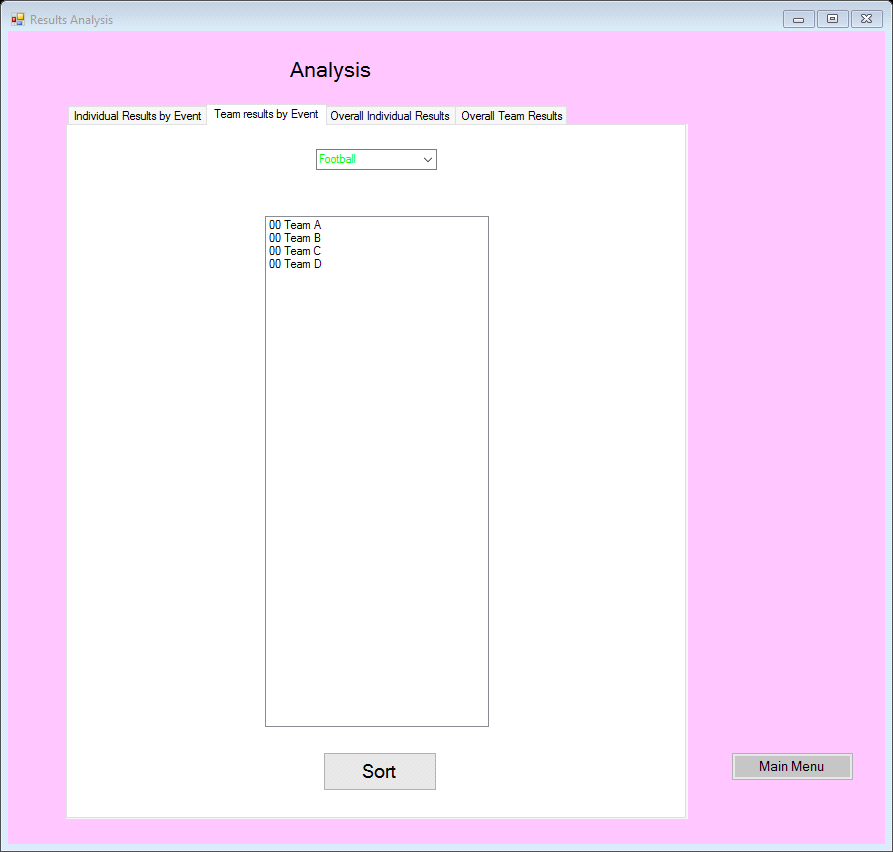
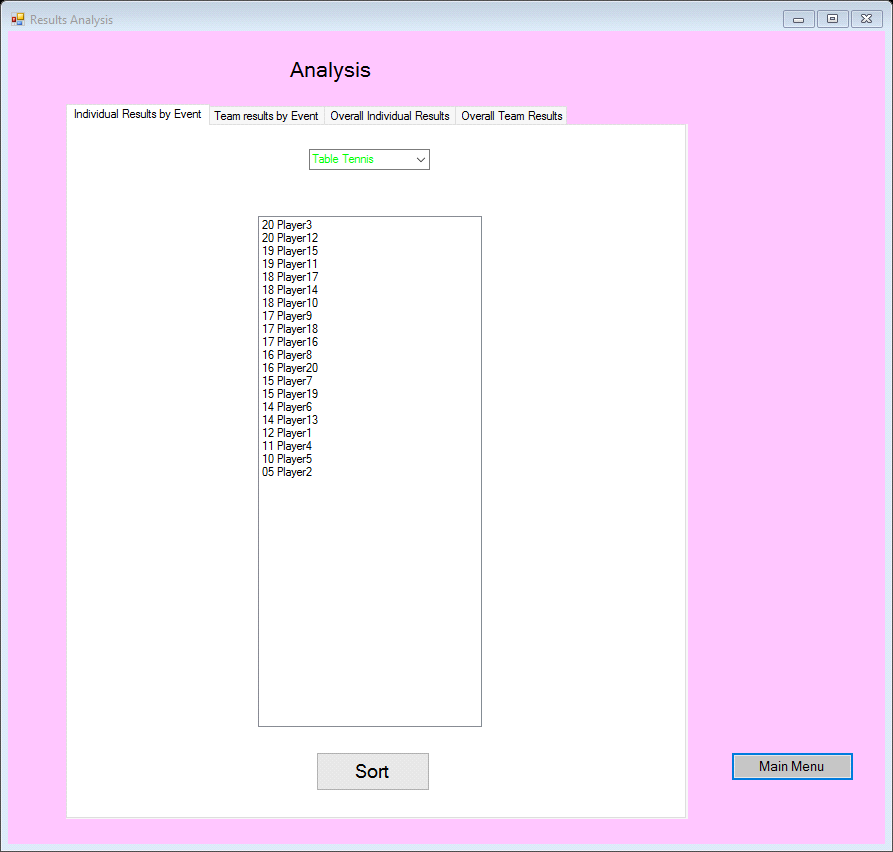
Private Sub btnMainMenu1\_Click(sender As Object, e As EventArgs) Handles btnMainMenu1.Click

Me.Hide()

Form1.Show()

End Sub

End Class



**frmRankings:**

Public Class frmRankings

Private Sub frmRankings\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

'Setup labels as an array

Dim playLabels As New List(Of Control)() From {

lbli1,

lbli2,

lbli3,

lbli4,

lbli5,

lbli6,

lbli7,

lbli8,

lbli9,

lbli10,

lbli11,

lbli12,

lbli13,

lbli14,

lbli15,

lbli16,

lbli17,

lbli18,

lbli19,

lbli20}

'Setup textboxes as an array

Dim textBoxes As New List(Of Control)() From {

txtInd1,

txtInd2,

txtInd3,

txtInd4,

txtInd5,

txtInd6,

txtInd7,

txtInd8,

txtInd9,

txtInd10,

txtInd11,

txtInd12,

txtInd13,

txtInd14,

txtInd15,

txtInd16,

txtInd17,

txtInd18,

txtInd19,

txtInd20}

'Adds labels with names of individual participants

For i As Integer = 0 To playLabels.Count - 1

playLabels(i).Text = ind(i).iName

Next i

'Adds individual events combobox

For i = 0 To 4

cboIndEvents.Items.Add(IEvent(i).EName)

Next i

'Adds team names to labels

lblt1.Text = team(0).TName

lblt2.Text = team(1).TName

lblt3.Text = team(2).TName

lblt4.Text = team(3).TName

'Adds team events to combo box

For i = 0 To 4

cboTeamEvents.Items.Add(TEvent(i).EName)

Next i

End Sub

Private Sub btnSaveInd\_Click(sender As Object, e As EventArgs) Handles btnSaveInd.Click

'Setup arrays

Dim playLabels As New List(Of Control)() From {

lbli1,

lbli2,

lbli3,

lbli4,

lbli5,

lbli6,

lbli7,

lbli8,

lbli9,

lbli10,

lbli11,

lbli12,

lbli13,

lbli14,

lbli15,

lbli16,

lbli17,

lbli18,

lbli19,

lbli20

}

'Setup arrays

Dim textBoxes As New List(Of Control)() From {

txtInd1,

txtInd2,

txtInd3,

txtInd4,

txtInd5,

txtInd6,

txtInd7,

txtInd8,

txtInd9,

txtInd10,

txtInd11,

txtInd12,

txtInd13,

txtInd14,

txtInd15,

txtInd16,

txtInd17,

txtInd18,

txtInd19,

txtInd20}

'Declare the variables

'i is a counter for the text boxes

Dim i As Integer

'Bool value to decide (currently false) until invalid number input

Dim xInvalid As Boolean

'Used to store each number as it is checked

Dim test As Integer

i = 0

xInvalid = False

'Loop the textbox values to check they are between 1 & 20

While i < textBoxes.Count And xInvalid = False

test = CInt(textBoxes(i).Text)

If test < 1 Or test > 20 Then

MsgBox("Please enter a number between 1 and 20" &

playLabels(i).Text & " is invalid",

MsgBoxStyle.Critical, "Error!")

xInvalid = True

End If

i += 1

End While

'Select case dependant what is visible in the combo box

'For loops to store results from textbox

If xInvalid = False Then

Select Case cboIndEvents.SelectedIndex

Case 0

For i = 0 To textBoxes.Count - 1

ind(i).iResult(0) = CInt(textBoxes(i).Text)

Next

Case 1

For i = 0 To textBoxes.Count - 1

ind(i).iResult(1) = CInt(textBoxes(i).Text)

Next

Case 2

For i = 0 To textBoxes.Count - 1

ind(i).iResult(2) = CInt(textBoxes(i).Text)

Next

Case 3

For i = 0 To textBoxes.Count - 1

ind(i).iResult(3) = CInt(textBoxes(i).Text)

Next

Case 4

For i = 0 To textBoxes.Count - 1

ind(i).iResult(4) = CInt(textBoxes(i).Text)

Next

End Select

End If

'Once results are saved, the text-boxes will go blank again to show they have saved.

For i = 0 To textBoxes.Count - 1

textBoxes(i).Text = ""

Next

End Sub

Private Sub btnSaveTeam\_Click(sender As Object, e As EventArgs) Handles btnSaveTeam.Click

'Setup arrays

Dim playLabels As New List(Of Control)() From {

lblt1,

lblt2,

lblt3,

lblt4}

Dim textBoxes As New List(Of Control)() From {

txtTeam1,

txtTeam2,

txtTeam3,

txtTeam4}

'Declare the variables

'i is a counter for the text boxes

Dim i As Integer

'Bool value to decide (currently false) until invalid number input

Dim xInvalid As Boolean

'Used to store each number as it is checked

Dim test As Integer

i = 0

xInvalid = False

'Loop the textbox values to check they are between 1 & 4

While i < textBoxes.Count And xInvalid = False

test = CInt(textBoxes(i).Text)

If test < 1 Or test > 4 Then

MsgBox("Please enter a number between 1 and 4" &

playLabels(i).Text & " is invalid",

MsgBoxStyle.Critical, "Error!")

xInvalid = True

End If

i += 1

End While

If xInvalid = False Then

'Select case dependant what is visible in the combo box

'For loops to store results from textbox

Select Case cboTeamEvents.SelectedIndex

Case 0

For i = 0 To textBoxes.Count - 1

team(i).TResult(0) = CInt(textBoxes(i).Text)

Next

Case 1

For i = 0 To textBoxes.Count - 1

team(i).TResult(1) = CInt(textBoxes(i).Text)

Next

Case 2

For i = 0 To textBoxes.Count - 1

team(i).TResult(2) = CInt(textBoxes(i).Text)

Next

Case 3

For i = 0 To textBoxes.Count - 1

team(i).TResult(3) = CInt(textBoxes(i).Text)

Next

Case 4

For i = 0 To textBoxes.Count - 1

team(i).TResult(4) = CInt(textBoxes(i).Text)

Next

End Select

End If

'Once results are saved, the text-boxes will go blank again to show they have saved.

For i = 0 To textBoxes.Count - 1

textBoxes(i).Text = ""

Next

End Sub

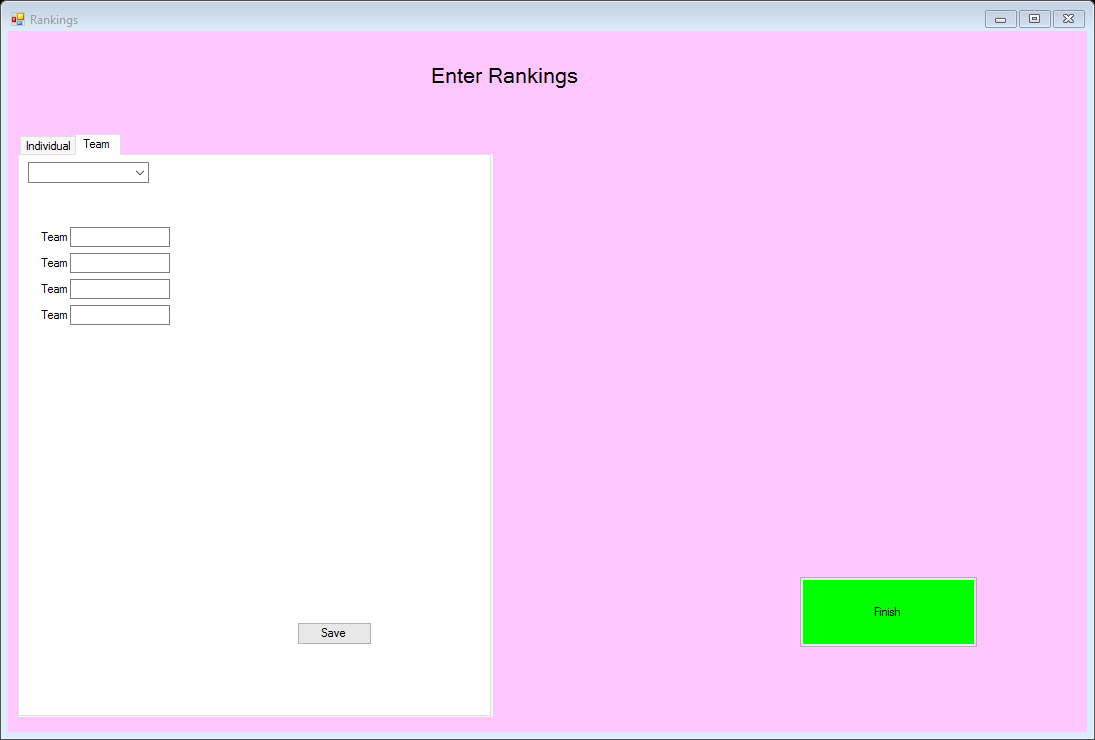
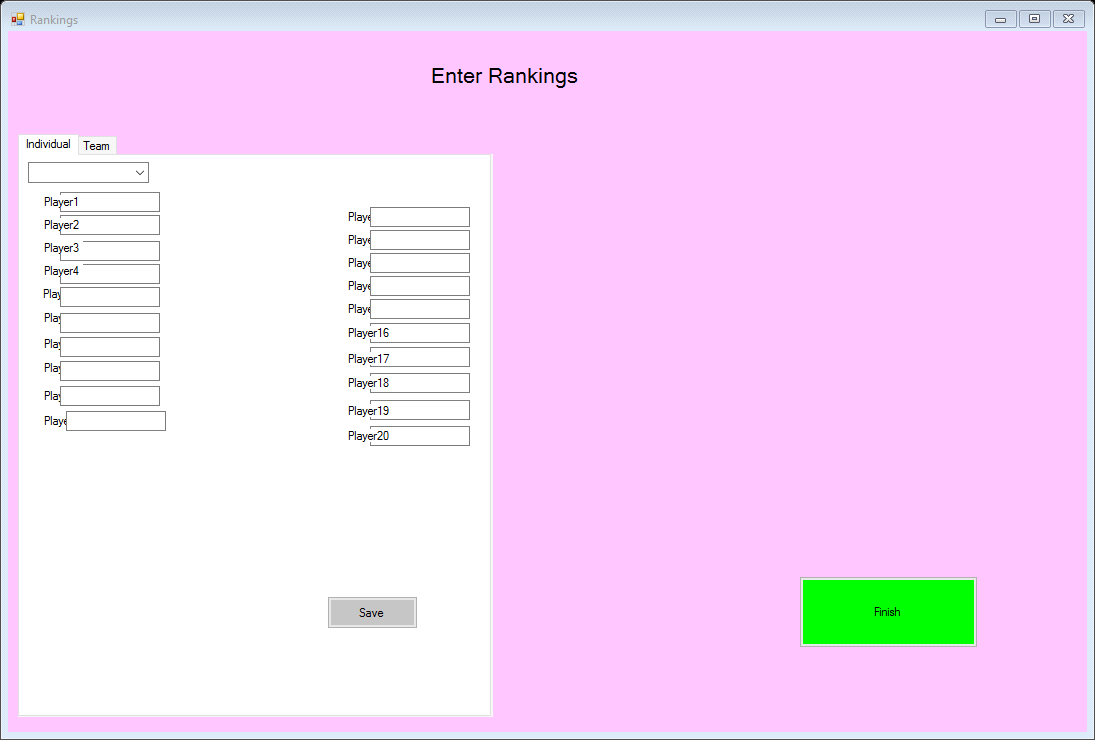
Private Sub btnMainMenu\_Click(sender As Object, e As EventArgs) Handles btnMainMenu.Click

Me.Close()

Form1.Show()

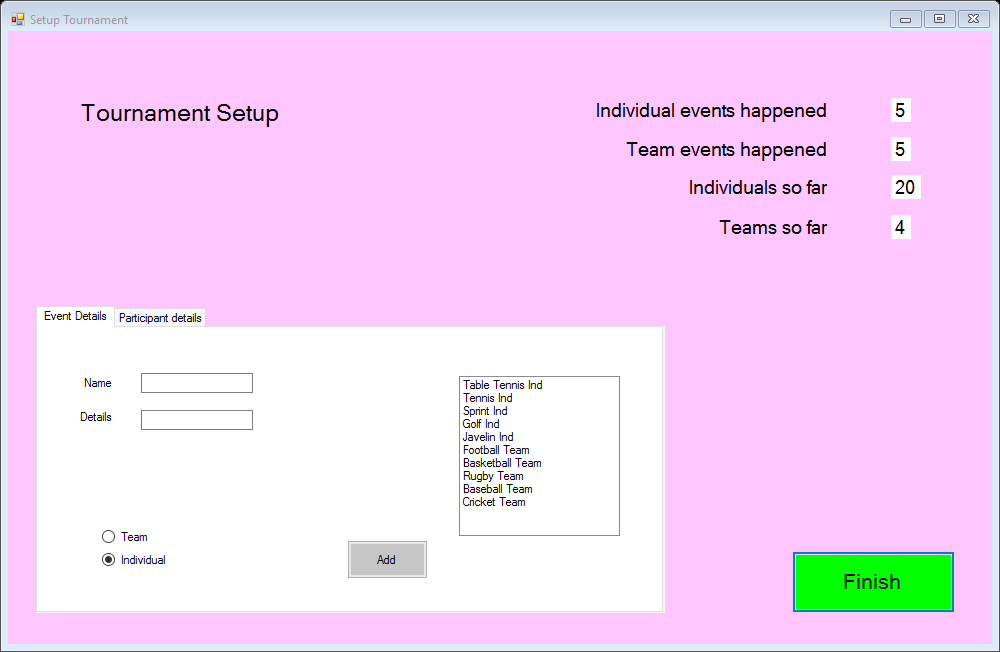
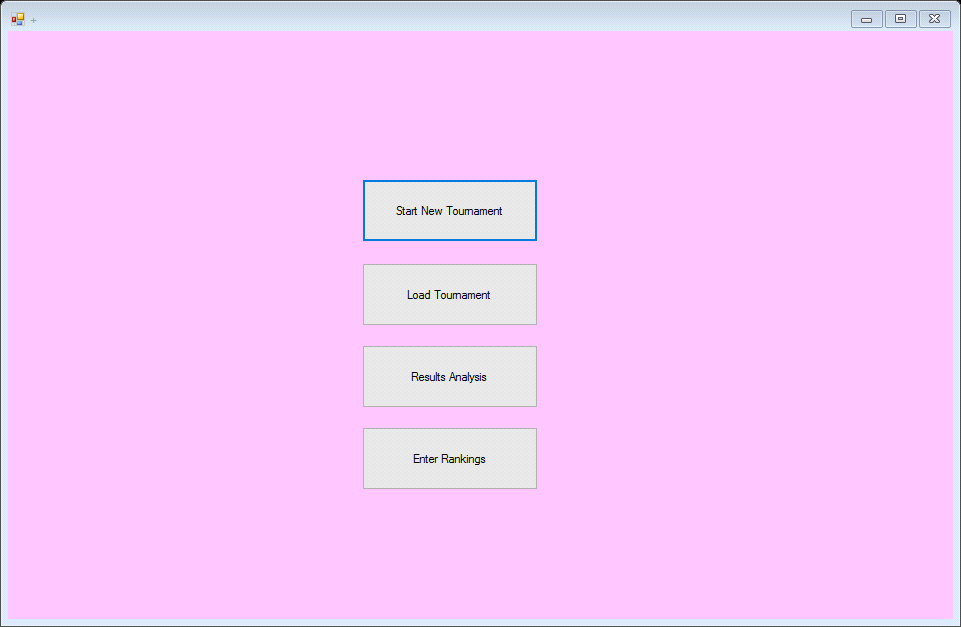
End Sub

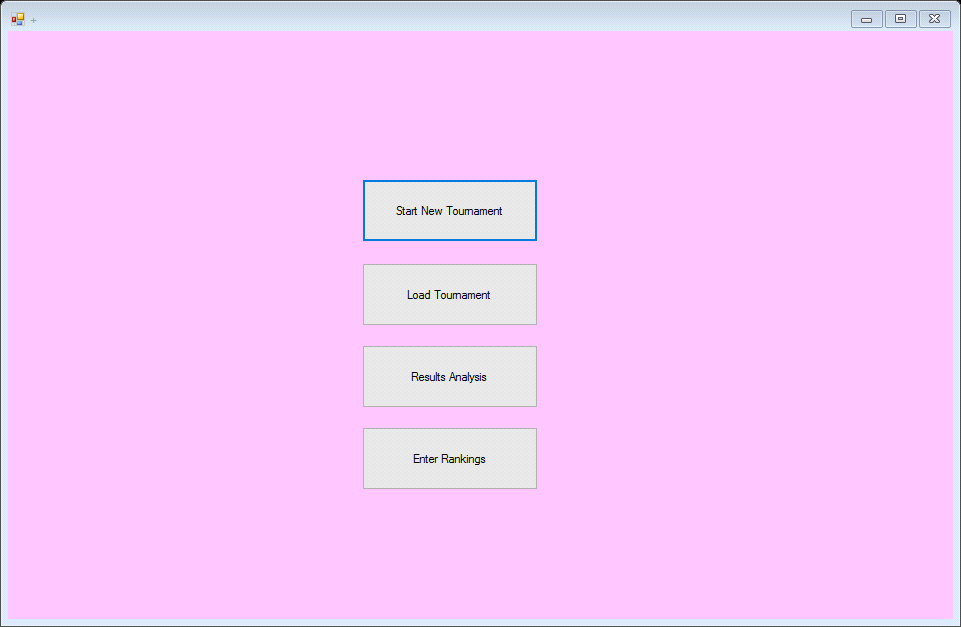
End Class

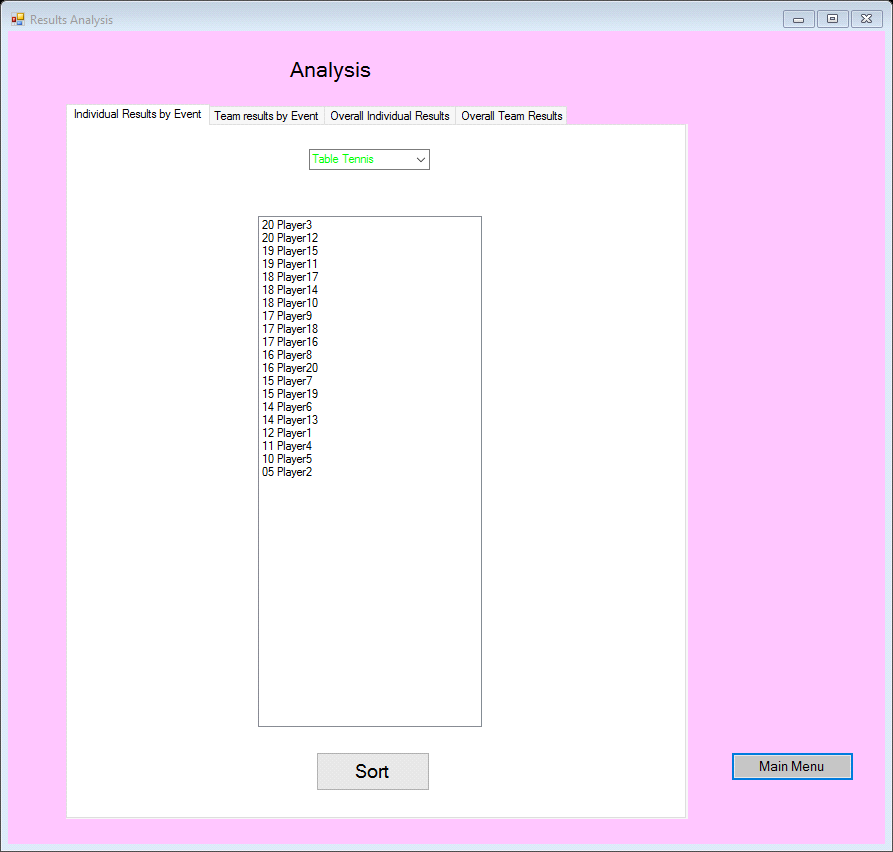
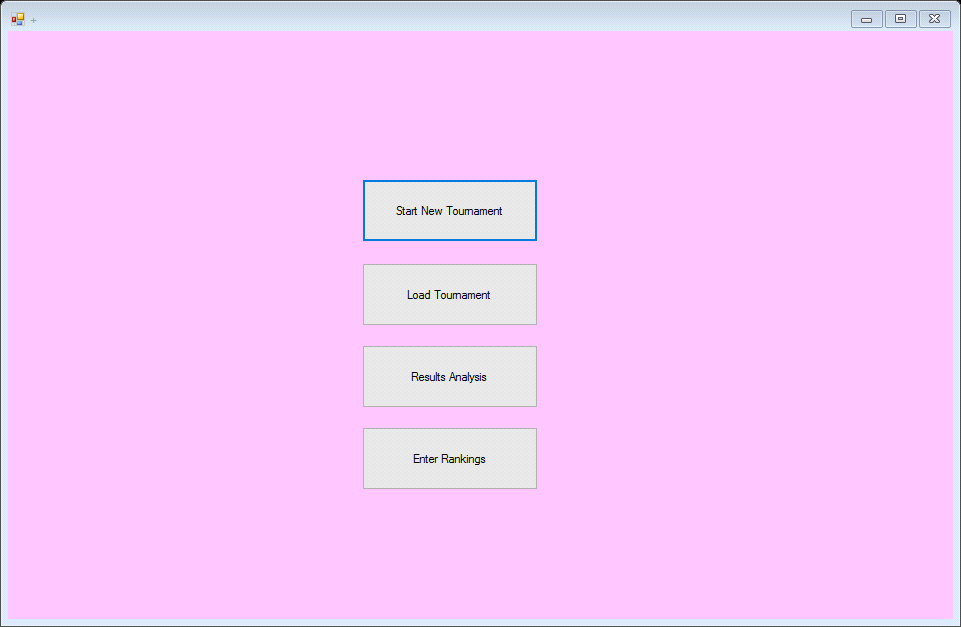


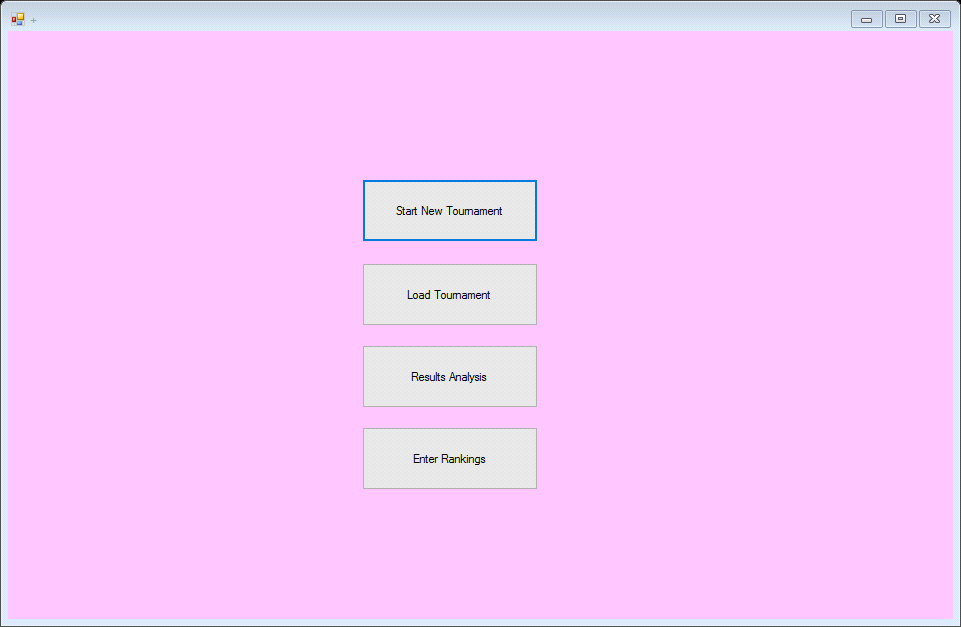
**Test Log**

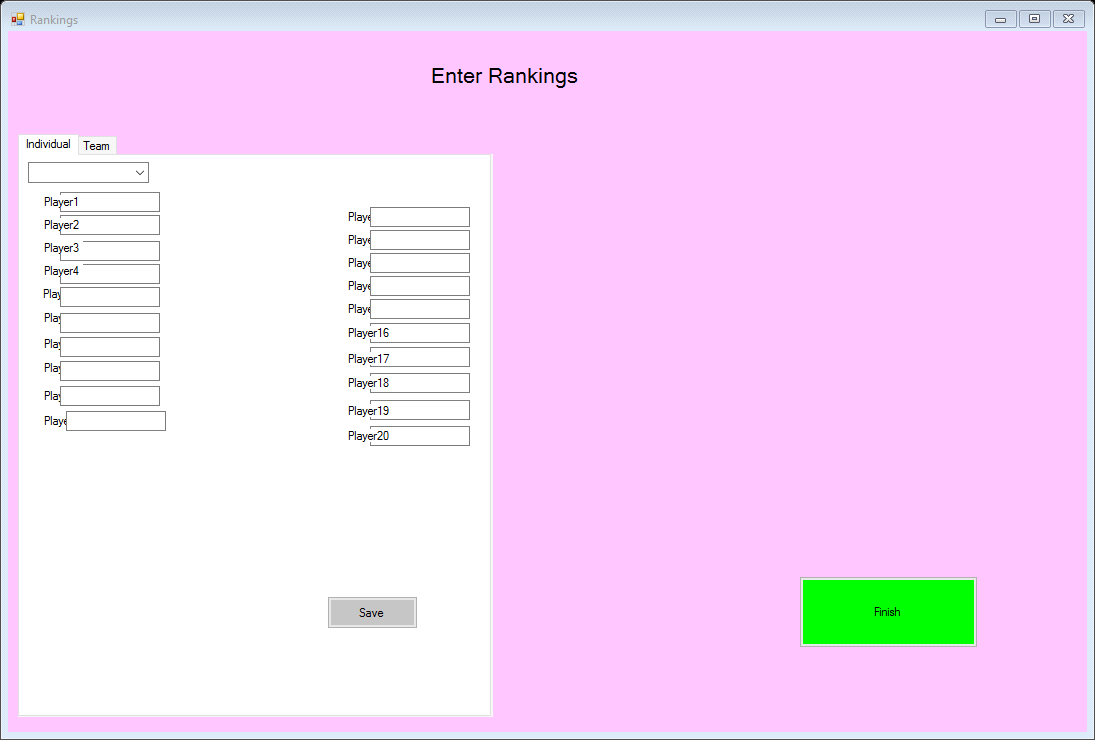
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Number** | **Name of Form** | **Type of Test (N,R,X)** | **Purpose**  **of Test** | **Test Data** | **Expected**  **Result** | **Actual**  **Result** | **Comments and Actions Taken** |
| 1 | frmMainMenu | N | To make sure the button works | BtnStartNew | The button will open the new form | It works, Setup form loaded | The button closes the form and opens the Start-up form |
| 2 | frmMainMenu | N | To make sure the button works | btnLoad | The button will open the new form | It works, Doesn’t Load | This is because there is not data ready to load meaning this works |
| 3 | frmMainMenu | N | To make sure the button works | btnResultsAnalysis | The button will open the new form | Works, Analysis form loads | The button closes the form and opens the Analysis form |
| 4 | frmMainMenu | N | To make sure the button works | btnEnterResults | The button will open the new form | Works, rankings form loads |  |
| 5 | frmSetup | N | To make sure the value changes | lblNumInEv | The value will change | Doesn’t work | In order for this to work it will need to have the value of the label set to the variable |
| 6 | frmSetup | N | To make sure the value changes | lblNumTeEv | The value will change | Works | The value changes when Team even it added |
| 7 | frmSetup | N | To make sure the value changes | lblIndSoFar | The value will change | Works | The value changes when Individual is added |
| 8 | frmSetup | N | To make sure the value changes | lblTeamsSoFar | The value will change | Works | The value changes when Team is added |
| 9 | frmSetup | N | To make sure the button takes the user back to the main menu | btnFinishSetup | The button will open a form | Works | The button works, it goes back to the Main Menu. The button also reads an error message if not enough events or players are listed. |
| 10 | frmSetup | N | To make sure the information for ‘individual Event’ is added to the list box | btnAddDet | The button will add the data to the list box | Works | The Event gets added to the list box |
| 11 | frmSetup | N | To make sure the information for ‘Team Event’ is added to the list box | btnAddDet | The button will add the data to the list box | Works |  |
| 12 | frmSetup | N | To make sure the information for ‘individual’ is added to the list box | btnAddPar | The button will add the data to the list box | Works |  |
| 13 | frmSetup | N | To make sure the information for ‘Team’ is added to the list box | btnAddPar | The button will add the data to the list box | Works |  |
| 14 | frmSetup | N | To make sure the Main Menu button works when pressed | btnMainMenu | The button will return to the Main Menu | Works |  |
| 15 | frmRankings | N | To make sure all of the data added is saved | btnSaveInd | The button will save the input data | Works |  |
| 16 | frmRankings | N | To make sure all of the added data is saved | btnSaveTeam | The button will save the input data | Works |  |
| 17 | frmSetup | N | To make sure an error occurs when there is missing data | btnAddDet | The button will cause an error to occur | Works |  |
| 18 | frmSetup | N | To make sure an error occurs when there is missing data | btnAddDet | The button will cause an error to occur | Works |  |
| 19 | frmSetup | N | To make sure an error occurs when there is too much data | btnAddPar | The button will cause an error to occur | Works |  |
| 20 | frmSetup | N | To make sure an error occurs when there is too much data | btnAddPar | The button will cause an error to occur | Works |  |

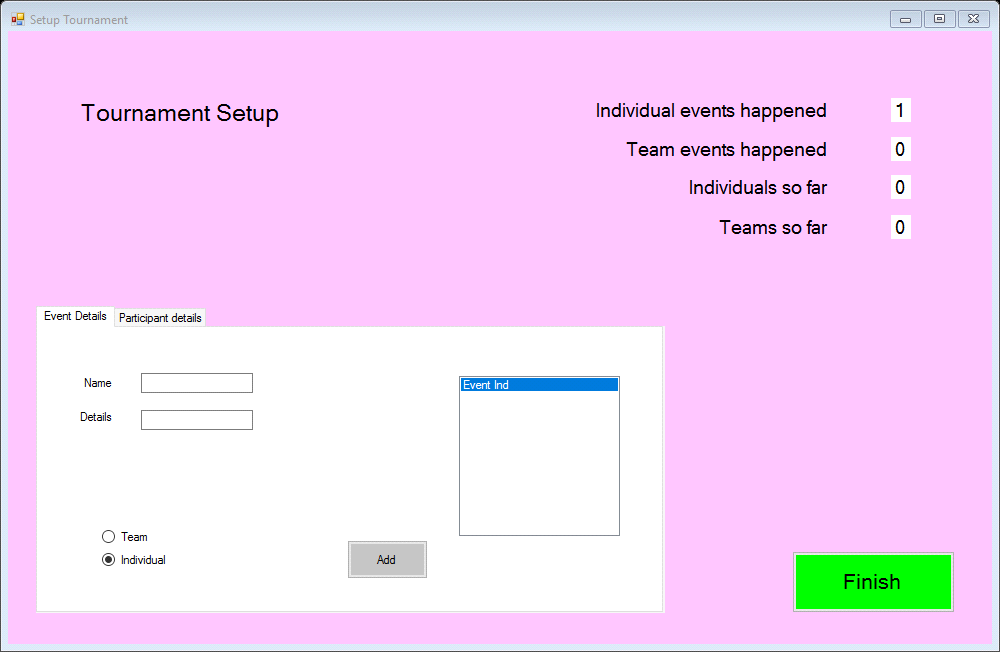
1.

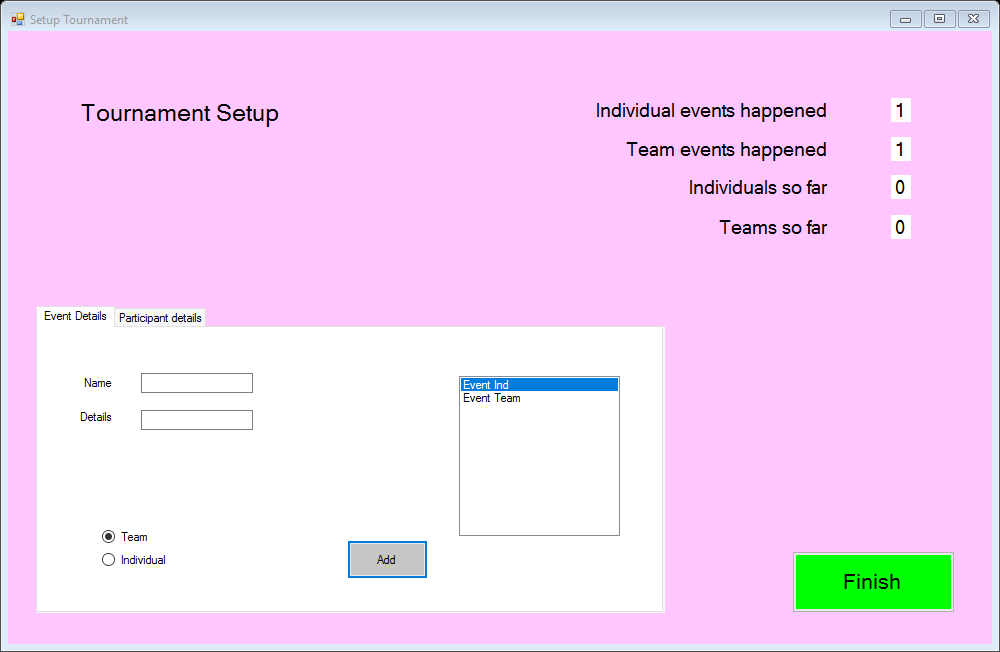
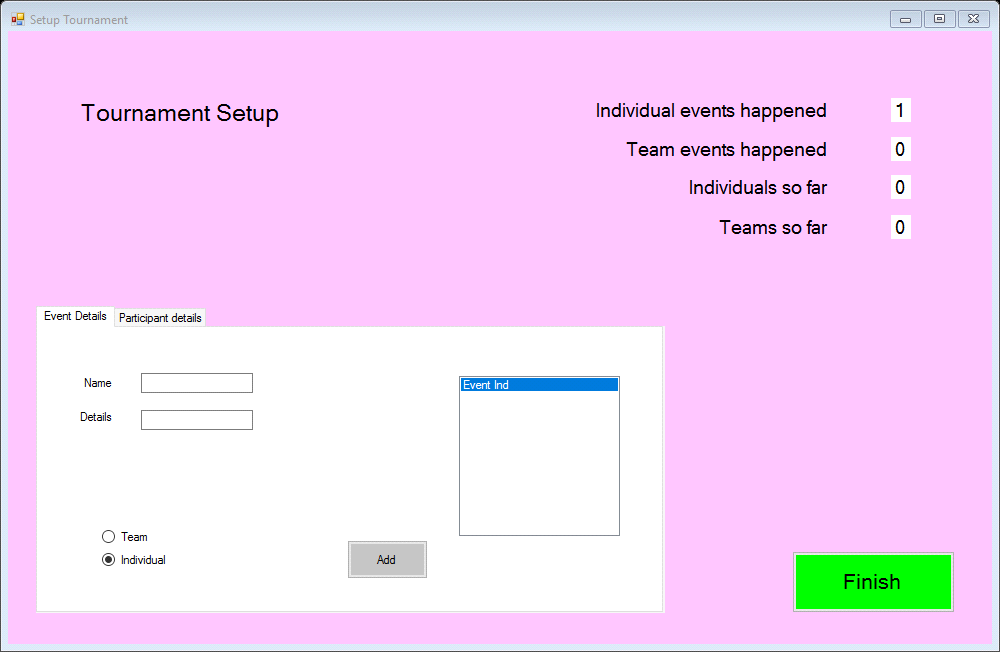
2. 

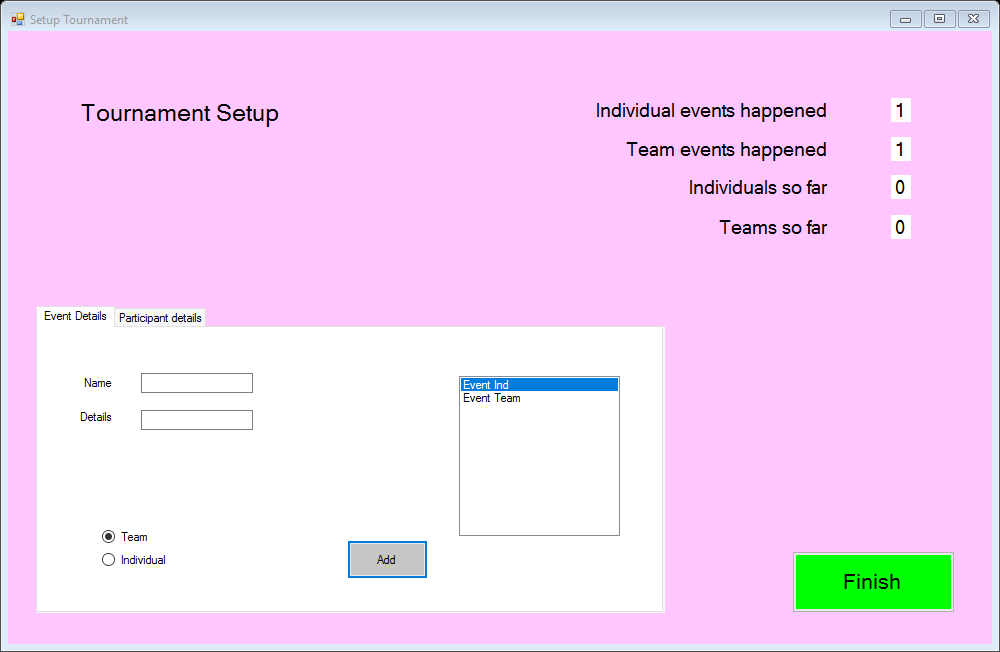
3. 

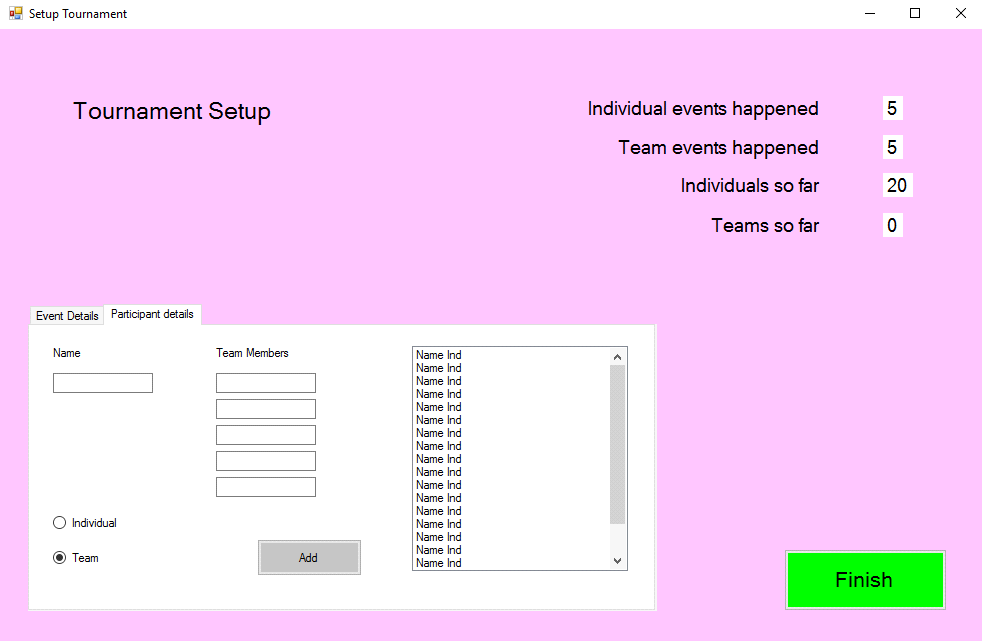
4. 

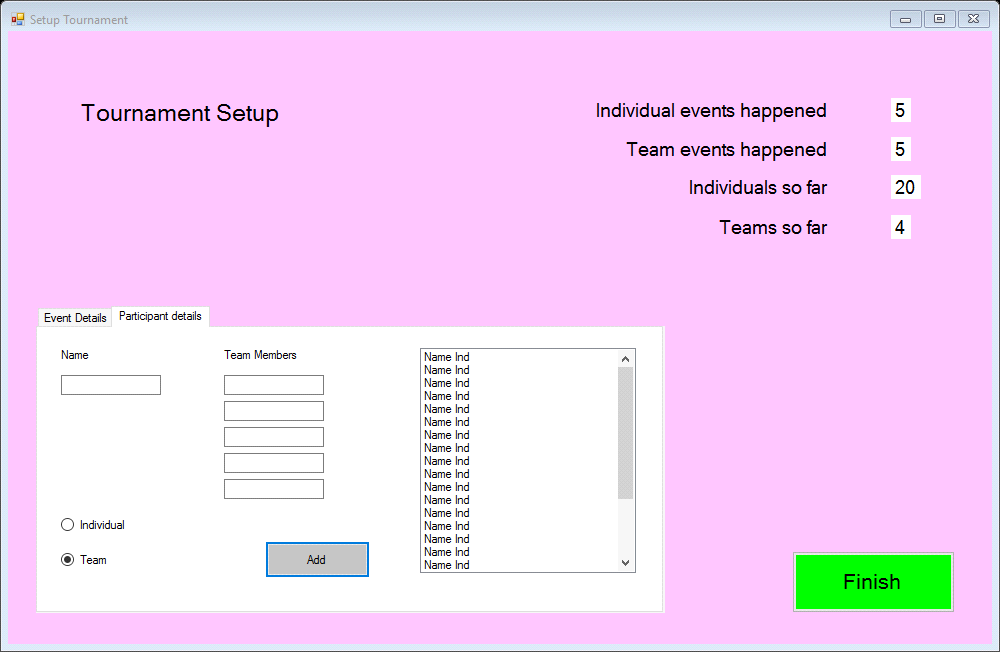
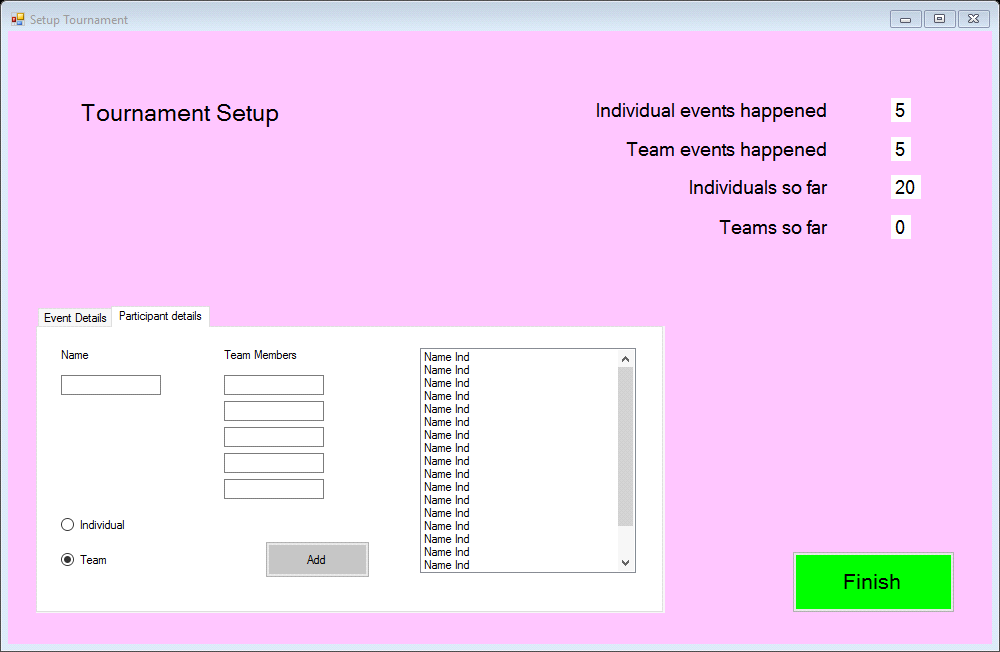


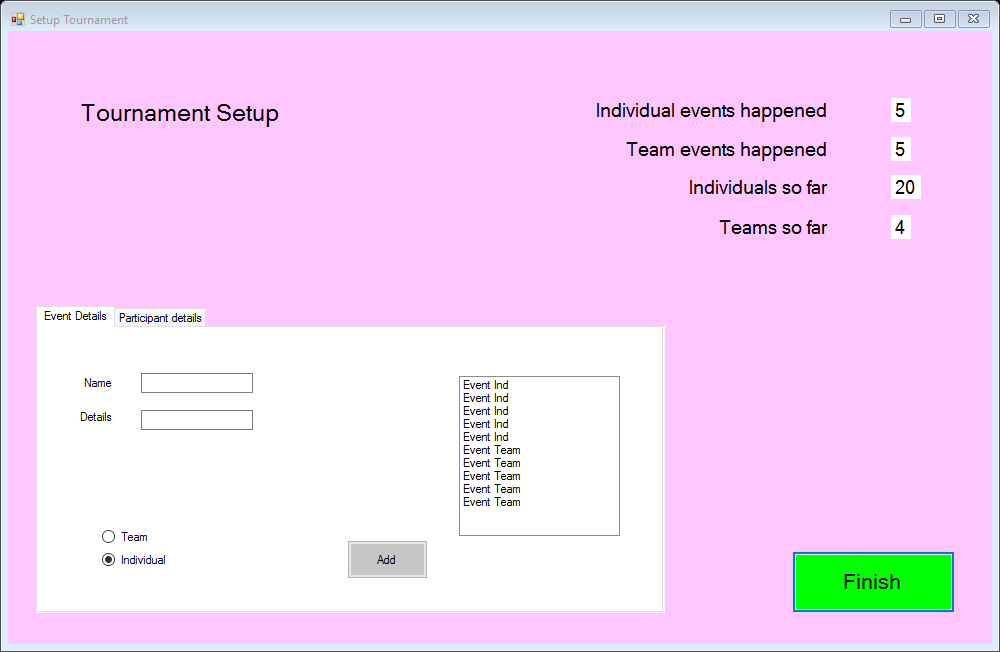
5. 

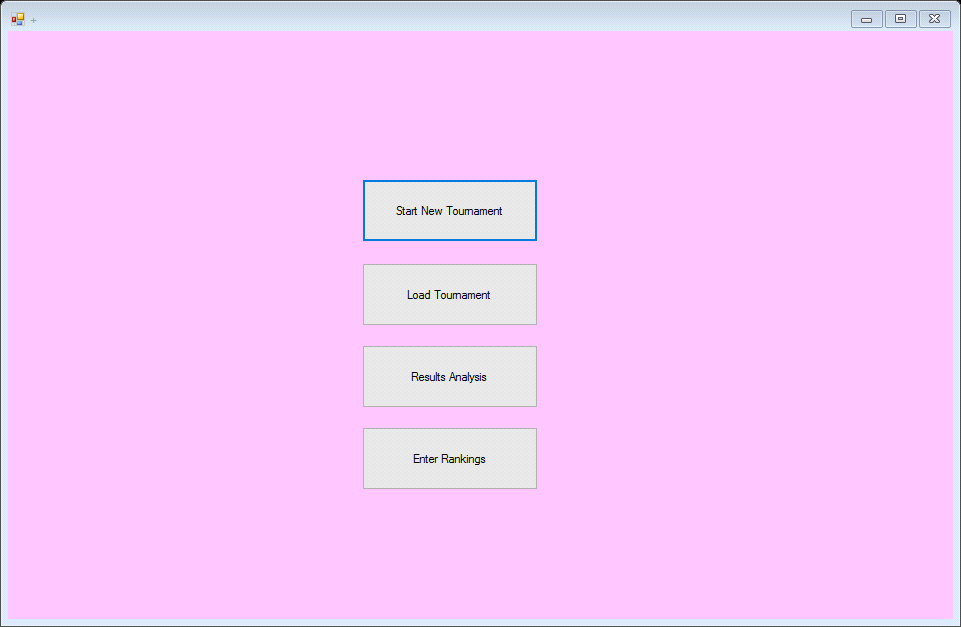
6. 

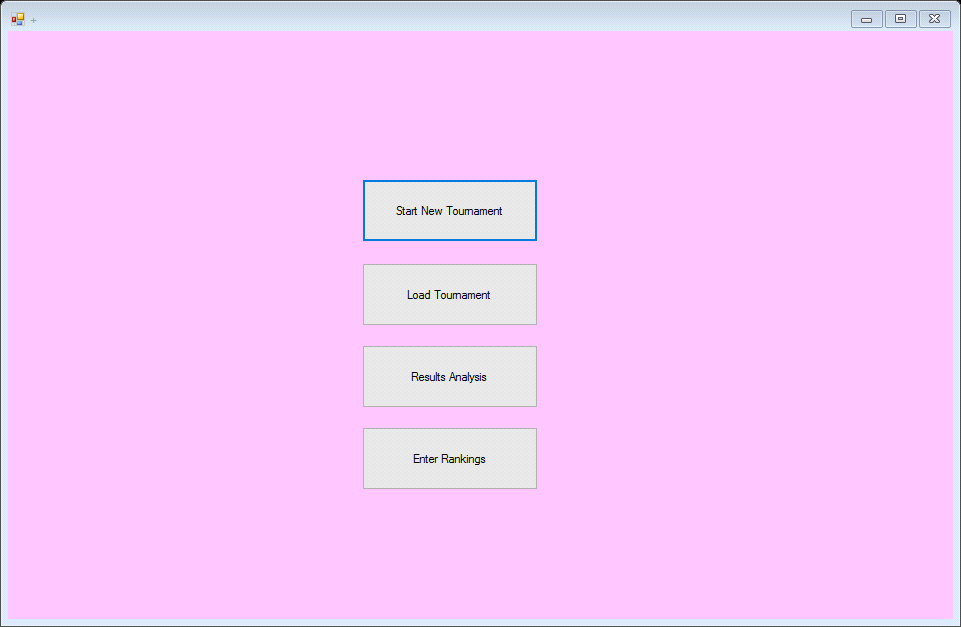
7. 

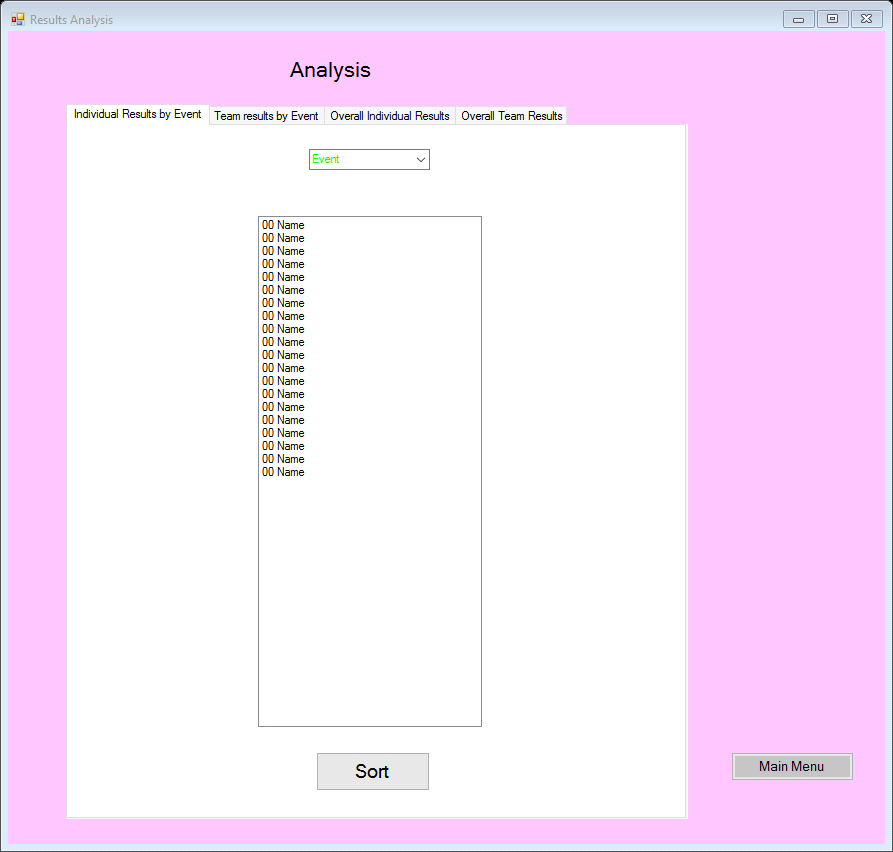


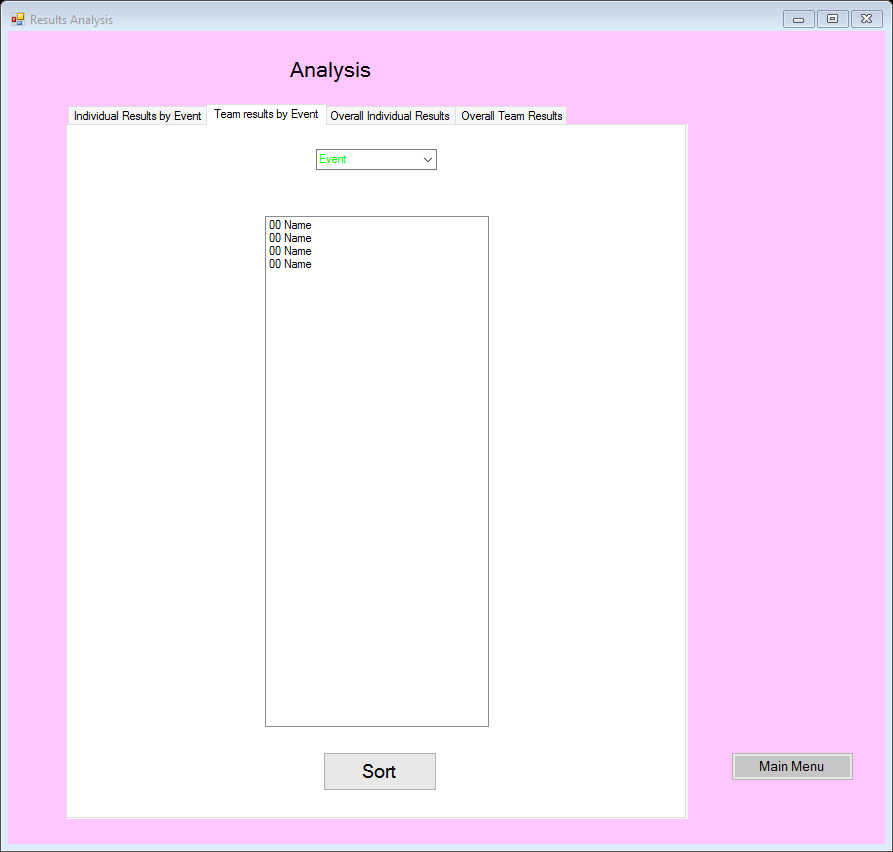
8.

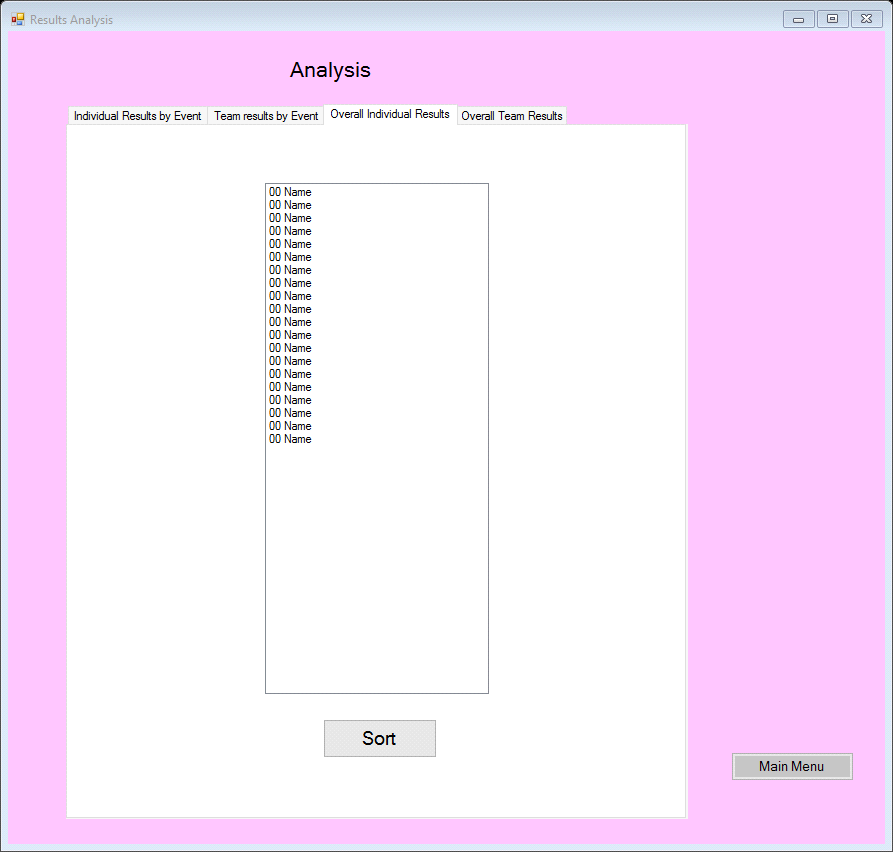
9. 

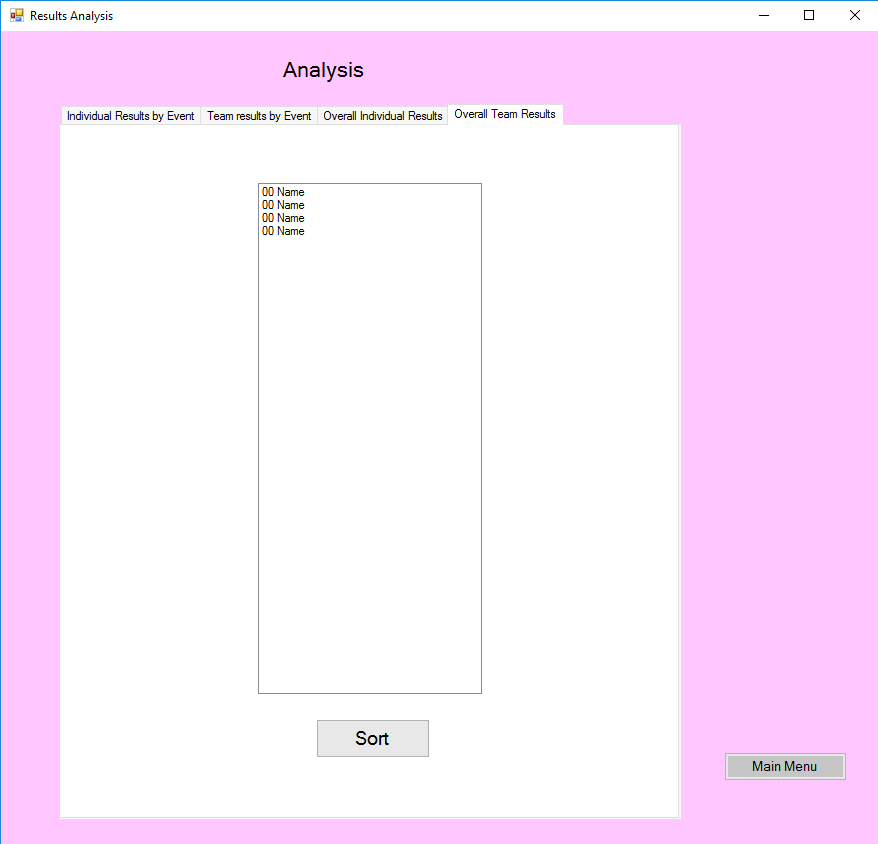


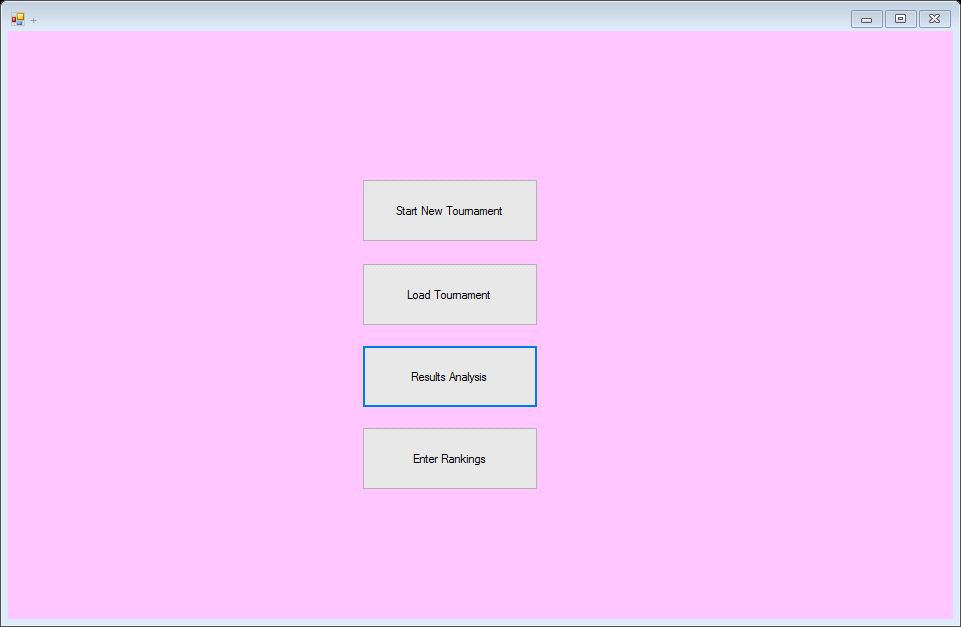
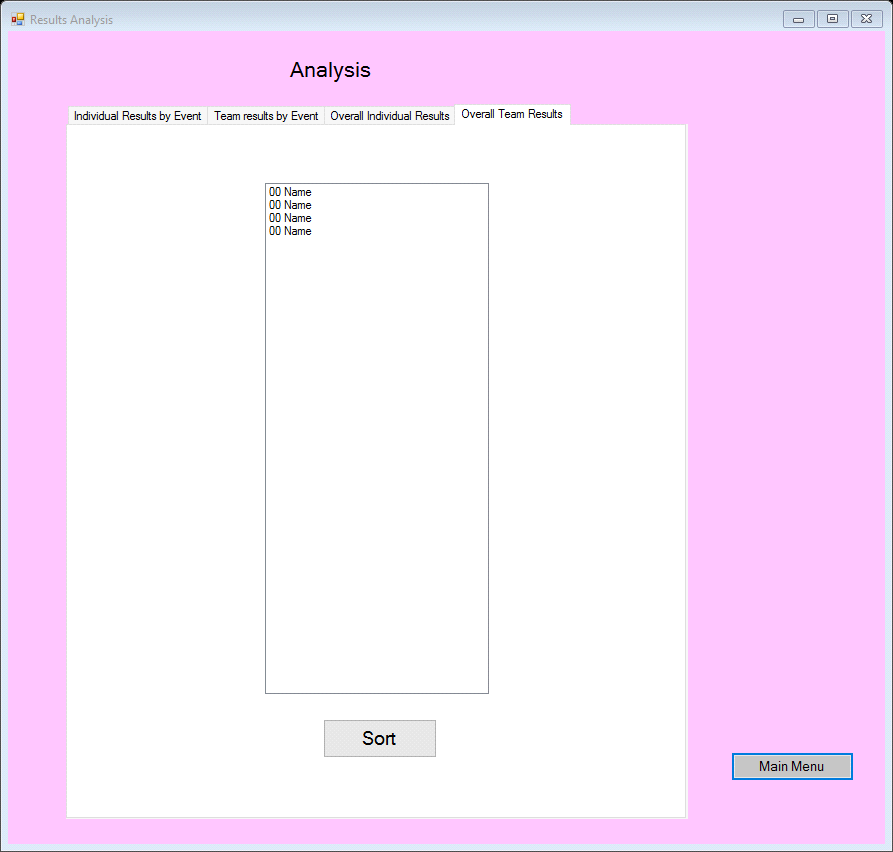
10. 

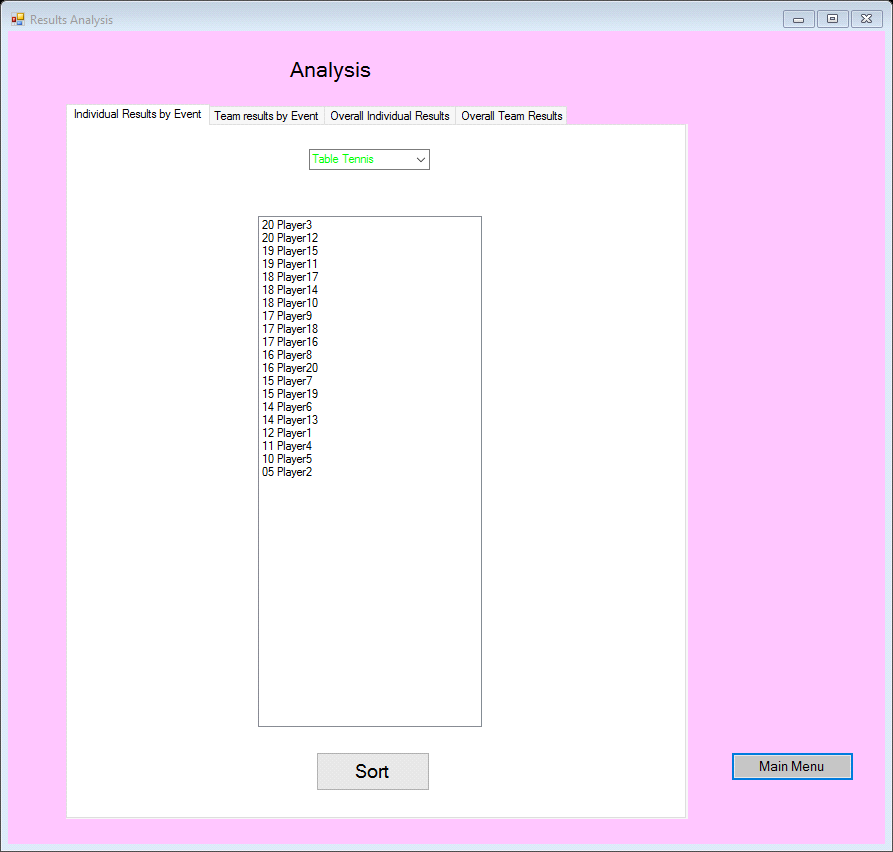


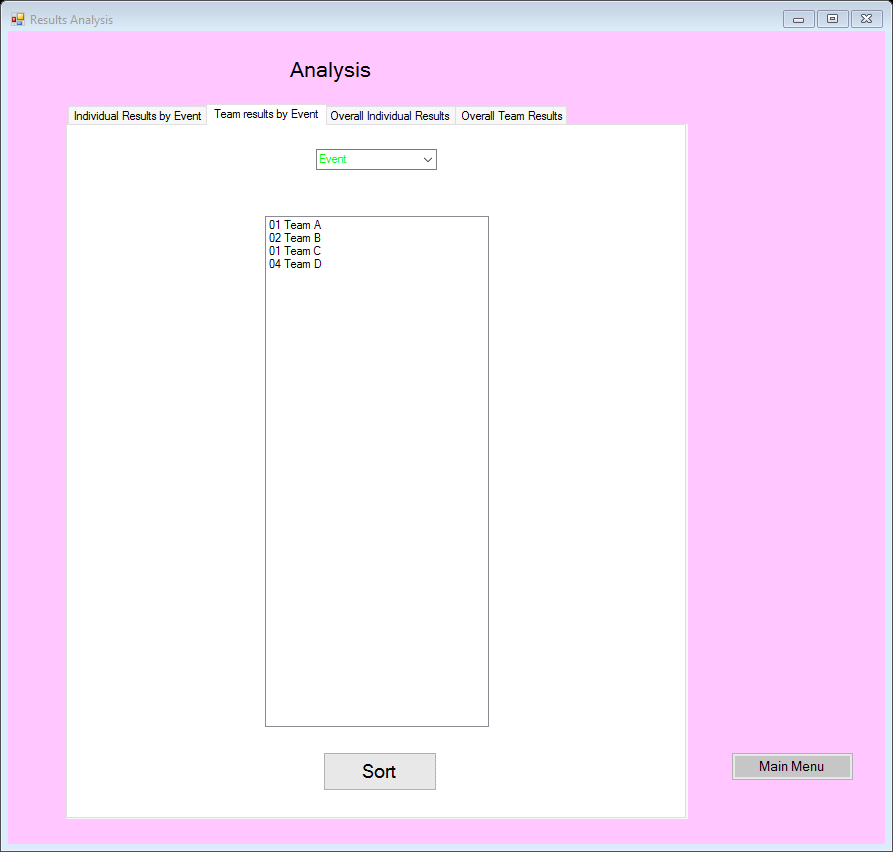
11. 

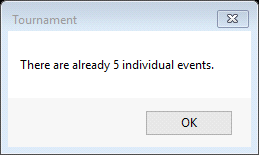
12. 

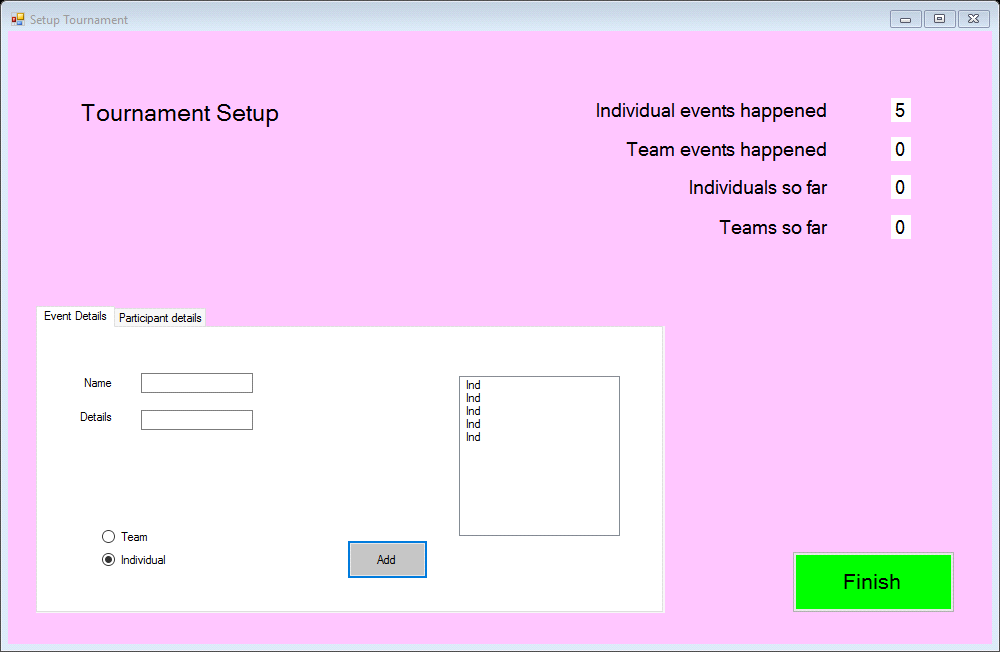
13. 

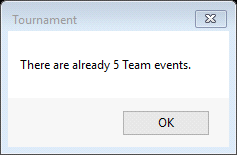
14. 

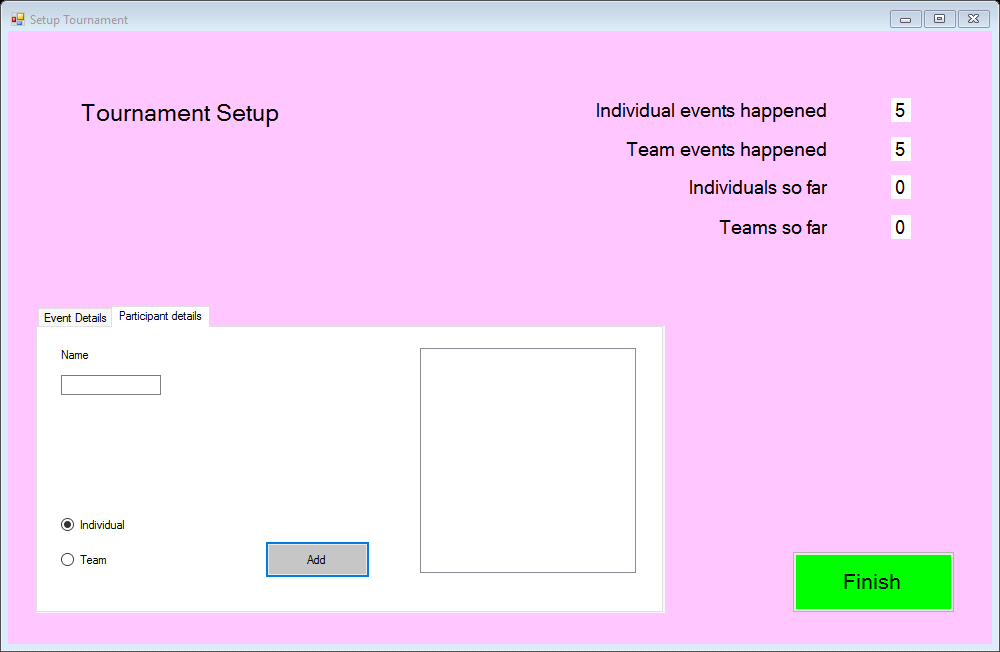
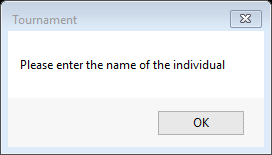
15. 

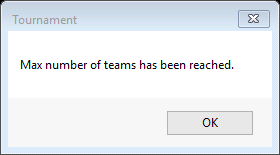
16. 

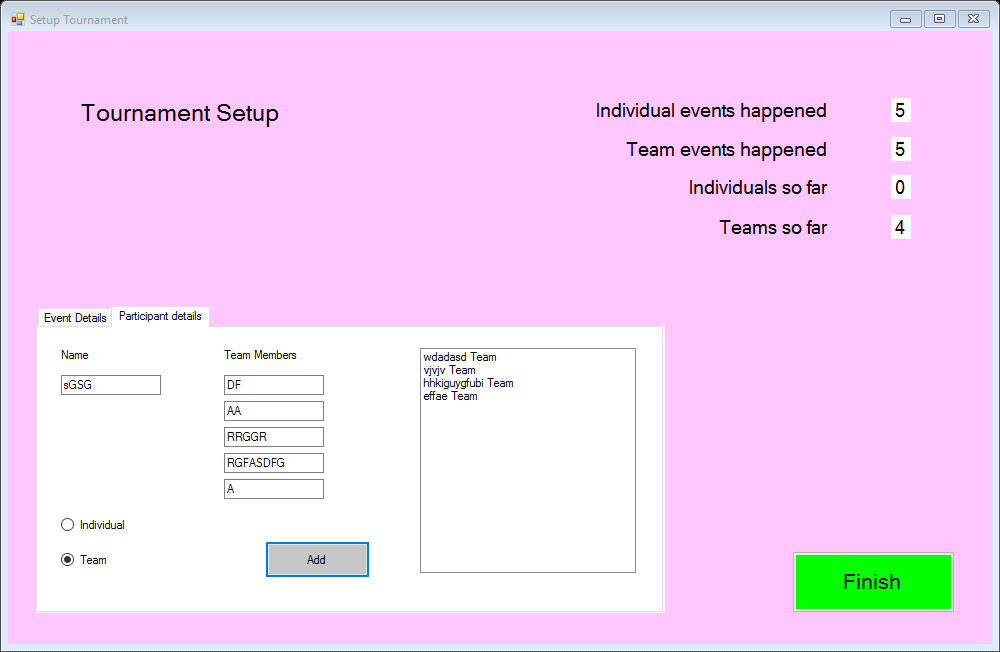


17. 

18. 

19. 



20. 

**User Review:**

When the program starts, it is very smooth and quick meaning that you can quickly access the buttons. The load button doesn’t do anything as it should because I haven’t created any data yet. The Enter results button doesn’t work, nor does the Rankings button which is good because there is not any data for the program to use. The Start button works and does entering all of the data. However, it takes a long time to enter the data. There are some validations which mean that you cannot enter too much data and for some text boxes you cannot enter no data. Once all of the data is created you can then access the enter results and analysis forms.

The program runs smoothly which is good for the user. The load button is disabled and the Enter results and analysis makes the program crash. The entering of data takes a long time, but works well with the analysis and entering results.

**Optimisation:**

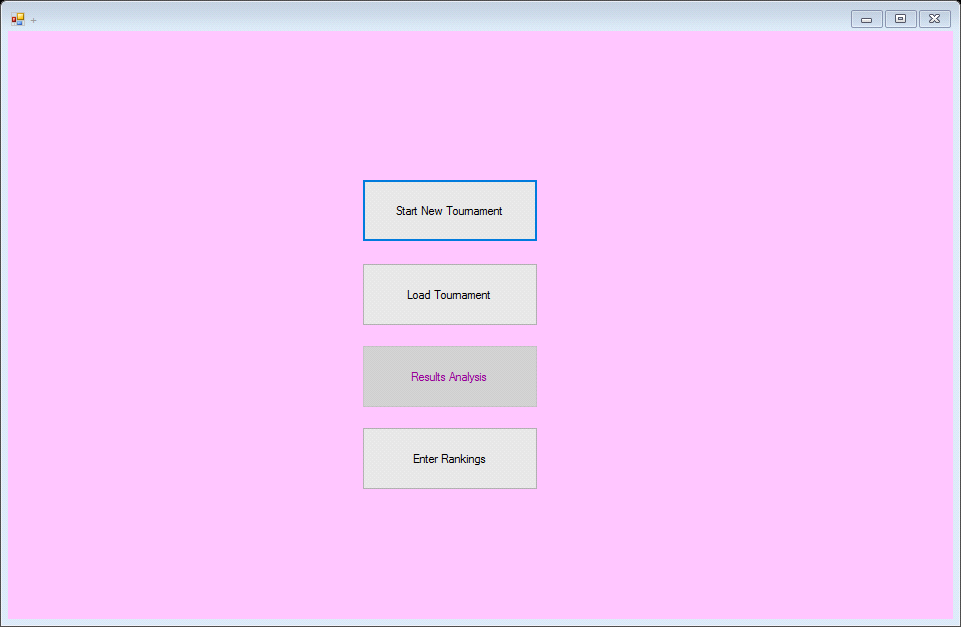
From the responses, I can tell that the users majorly like this program, however there are some results that need to be made. They liked the speed of the program, the quick navigation. However, they did not like when the program crashed due to missing information so I will need to disable to button and they also did not like that it took so long to enter the data so I will be planning on adding the Enter key to input data quicker.

The first improvement I made was the for there to be a default selected option on the setup page. In order to do this, I had to select Checked = True





The second improvement I made is for the Analysis and Enter Results buttons to be disabled to prevent them crashing the program whilst they do not have any data.



The other improvement I am making is that instead of the user constantly navigating to select the textbox and then pressing ‘add’, they will be able to simply press the ‘Enter’ Key to input details if there are any.

Private Sub KeyIsDown(sender As Object, e As KeyEventArgs) Handles MyBase.KeyDown

If e.KeyCode = Keys.Enter Then

'Checks for the option individual selected

If optIndividual.Checked = True Then

'Checks if there is enough room in the listbox for another event

If IEventCount < 5 Then

IEvent(IEventCount).EName = txtEventName.Text

'input for event name

IEvent(IEventCount).EDesc = txtEventDesc.Text

'input for description

IEventCount += 1

'adds 1

lblNumInEv.Text = IEventCount

'adds 1

lstEventsAdded.Items.Add(txtEventName.Text & " Ind")

'Adds the item to the listbox

Else

'Makes the user aware there are too many events added

MsgBox("There are already 5 individual events.")

End If

End If

'Checks for the option team selected

If optTeam.Checked = True Then

'Checks if there is enough room in the listbox for another event

If TEventCount < 5 Then

TEvent(TEventCount).EName = txtEventName.Text

'input for event name

TEvent(TEventCount).EDesc = txtEventDesc.Text

'input for description

TEventCount += 1

'adds 1

lblNumTeEv.Text = TEventCount

'Adds the item to the listbox

lstEventsAdded.Items.Add(txtEventName.Text & " Team")

'Makes the user aware there are too many events added

Else

MsgBox("There are already 5 Team events.")

End If

End If

'Checks that the details are correctly completed

If txtPartName.Text <> "" Then

If optIndPar.Checked = True Then

If indCount < 20 Then

ind(indCount).iName = txtPartName.Text

ind(indCount).iResult = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}

indCount += 1

lblIndSoFar.Text = indCount

lstPartAdded.Items.Add(txtPartName.Text & " Ind")

Else

MsgBox("Max number of individual players has been reached.")

End If

End If

'checks the radio button for Team selected

If optTeamPar.Checked = True Then

'Asks for 5 names

If txtTeamMem1.Text = "" Or txtTeamMem2.Text = "" Or txtTeamMem3.Text = "" Or txtTeamMem4.Text = "" Or txtTeamMem5.Text = "" Then

MsgBox("Please enter 5 names for the team")

Else

'Checks that there are less than 4 teams

If teamCount < 4 Then

'Team name

team(teamCount).TName = txtPartName.Text

'Team members

team(teamCount).TPlayerName = {txtTeamMem1.Text, txtTeamMem2.Text, txtTeamMem3.Text, txtTeamMem4.Text, txtTeamMem5.Text}

'Increases the team count since a team has been added.

team(teamCount).TResult = {0, 0, 0, 0, 0}

teamCount += 1

lblTeamsSoFar.Text = teamCount

'Adds the team to the listbox

lstPartAdded.Items.Add(txtPartName.Text & " Team")

Else

'Personalised message for when max num teams been added

MsgBox("Max number of teams has been reached.")

End If

End If

End If

Else

MsgBox("Please enter the name of the individual")

End If

'Clears textbox for teamn/player name

txtPartName.Text = ""

'Clear textboxes for player names

txtTeamMem1.Text = ""

txtTeamMem2.Text = ""

txtTeamMem3.Text = ""

txtTeamMem4.Text = ""

txtTeamMem5.Text = ""

End If

End Sub

I have also created a way for the results of the data to be saved in a file:

Private Sub btnSaveIR\_Click(sender As Object, e As EventArgs) Handles btnSaveIR.Click

'Declaring variable

Dim sb As New System.Text.StringBuilder()

'Loops through items in the listbox

For Each o In lstIndRes.Items

sb.AppendLine(o)

Next

'If user selects okay, the file will be written

SaveFileDialog1.Filter = "TXT Files (\*.txt\*)|\*.txt"

If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK \_

Then

My.Computer.FileSystem.WriteAllText \_

(SaveFileDialog1.FileName, sb.ToString(), True)

End If

MsgBox(SaveFileDialog1.FileName & " The file has been saved successfully")

End Sub

Private Sub btnSaveTR\_Click(sender As Object, e As EventArgs) Handles btnSaveTR.Click

'Declaring variable

Dim sb As New System.Text.StringBuilder()

'Loops through items in the listbox

For Each o In lstTeamRes.Items

sb.AppendLine(o)

Next

'If user selects okay, the file will be written

SaveFileDialog1.Filter = "TXT Files (\*.txt\*)|\*.txt"

If SaveFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK \_

Then

My.Computer.FileSystem.WriteAllText \_

(SaveFileDialog1.FileName, sb.ToString(), True)

End If

MsgBox(SaveFileDialog1.FileName & " The file has been saved successfully")

End Sub

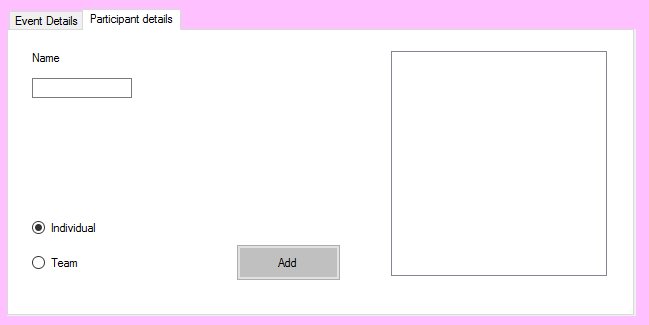
**Evaluation:**

Design Process:

For this, I had to take in and interpret how I would make the program meet the following targets:

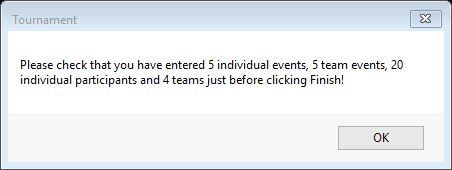
**The users MUST be able to enter themselves in a team or individual.**

Which they can do as I have developed a radio button selection that tailors their input options dependent upon whether they choose to be an individual or a team.

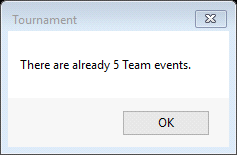


**The users will be in 4 teams of 5 and 20 individuals**

This has been completed as I have not allowed the user to progress unless they reach both of these amounts.



**Each team or individual will complete 5 events.**



The user cannot advance until they have entered 5 events and they can enter no more. This is good because it meets the user requirements.

**Each event will be either team of individual.**

For this, I have created two buttons that will allow the user to select whether to input individuals as a part of a team, or as themselves. This is useful because then the user will be able to decide whether a player belongs to a team or is an individual.

**The events will vary from sporting to academic challenges.**

For this program, I had only entered sporting activities, but to improve it I could append some of those activities to academic ones so that the player-base could increase since there are activities that apply to a wider range of students.

**Individuals and teams will be awarded points dependant on their rank in the events.**

Yes, however this is down to what their points are entered as, but when sorted they will be listed in their sorted order. So to improve this

**The points can be decided by us as they have not yet been decided.**

There is a form that allows the players' points to be entered. This is good because then this allows the events to be judged after they are played. There is also a pre-set of data that can be used.

**The college would like the ability for a single event to be entered on its own.**

This is possible as there is a section on the analysis form that allows points to be viewed by event.

Personal Self-Management:

For this, I had to learn how to program in Visual Basic, which mean there would be a lot of writing and appending code since I had to learn from scratch. It is a simple bit of knowledge to grasp after a lot of trying and checking bus is quite easy after lots of code has been written. For this project, I had to be quite strict, as I did not have a variable time to develop the program, meaning that I had to spend a lot of time writing and learning the code so that I could get it right since the designing process was also quite long. I had to create some initial designs and then improve them also finding the time to have them reviewed which meant I would have less time to actually write the program. I did find a lot of errors when writing the code they were minor since all the errors only really consisted of the names of the objects being misspelled of wrong completely.