YourFitnessFriend

Zayd Hussain

Advanced Higher

Computing Science

Project

Documentation

Index:

Analysis....3
Description of the Problem....3
Scope....4
Constraints....5
Boundaries....5
UML: Use Case Diagram....6
Requirements Specification....7
Project Plan....8
List of Resources....10
Design....11
Pseudocode....11
UML Class Diagram....18
Data Dictionary (Database Integration)....19
Query Design....21
Entity Relationship Diagram....22
User Interface (Wireframes)....23
Implementation....27
Software Design and Development User Interface....27
Database Design and Development Implemented Tables....30
Software Design and Development Program Code....32
Database Design and Development SQL Code....47
Research and Development of New Skills and Knowledge....50
On Going Testing....52
Log of Ongoing Testing Errors....53
Testing the Solution....55
Final Test Plan....55
Persona....55
Test Table....56
Requirements Testing....59
Test Results....75
Evaluation of the solution....76
Fitness for Purpose....76
Requirements Check List....77
Maintainability....78
Robustness....81
Final Evaluation....83

Analysis

Description of the Problem

I intend to make a fitness tracker that will track the end user's fitness, calculate a fitness goal and provide an action plan which includes a diet plan depending on their actual fitness goal. The program takes in the users' data and will use this to calculate their BMR and then a calorie deficit or a calorie surplus depending on the user's goals. The program will also show diet plans depending on their end goal and will create an action plan for them. To calculate the user's BMR I will be using the Harris-Benedict formula to do this. I will be implementing Software Design and Development and will executing my project by using Object-Oriented Programming which will carry out an array of Objects and Records which will store the data for the user's fitness goals. My main features will include a login system containing a register account page and login page which will direct the user to the home page. The homepage will contain a menu which will direct the user to a page where they can enter their data by using their username. This will display a unique ID to the user so they can view their action plan which include their goal, action steps and diet plan. The home page will also include a help section and include advice on what the user needs assisting with regards to their health. When the user is finished, they can also logout of the program if they are done.

My projects meets all of the Advanced Higher Computing requirements because it will have an easy to use user interface, it will be maintainable and also robust. It will also meet all of the Advanced Higher Computing requirements as my project integrates with a Database using MySQL for the login system. It will store all the user login details in the database and will also store the fitness data. My project will make use of a standard algorithm and I will use an insertion sort algorithm as I believe it will be the best way to organise the fitness data and then display that to the user. I have made sure that my insertion sort will sort through an array of records and specifically sort through the fitness goals to ensure I run into no errors.

Scope

The Scope of my project will include:

1. A full deep analysis on my project detailing things such as a description of my program. It will also go into detail about Advanced Higher concepts and integration, scope, boundaries and any constraints use case diagram, requirements and a fully detailed project plan.
2. A completed design including:
    - Pseudocode for my program to design the data flow and enhancements of the Advanced Higher concepts.
    - A Complete data dictionary for my tables in my database.
    - List of MySQL Queries Designs that I will be using in my program.
    - Wireframe designs for all of my User Interfaces including all inputs, validation and with annotations. All showing the intended user interface for my fitness tracker.
    - Class Diagram for Array of Objects showing Class Name, Encapsulation, Data Types, Constructor, Properties and Methods.
3. A well-designed and successfully implemented fitness tracker with a functional login system and will allow users to track their fitness goals. With all of my project code showing. I will also include Ongoing Testing during implementation showing Research and Development of any New Skills or Knowledge during my Implementation.
4. A finalised test plan with a summary of final testing including a persona, test cases, test results and requirements testing which will show all evidence of testing including screenshots of my program and screenshots of database output tables generated by the MySQL statements.
5. An evaluation report which will evaluate my program in terms of fitness for purpose. Showing how closely my program matches the functional requirements and how my results of my testing show this. It will also how the maintainability and robustness of my program

Constraints

When it comes to my constraints for my project there are a few factors that apply for this project:

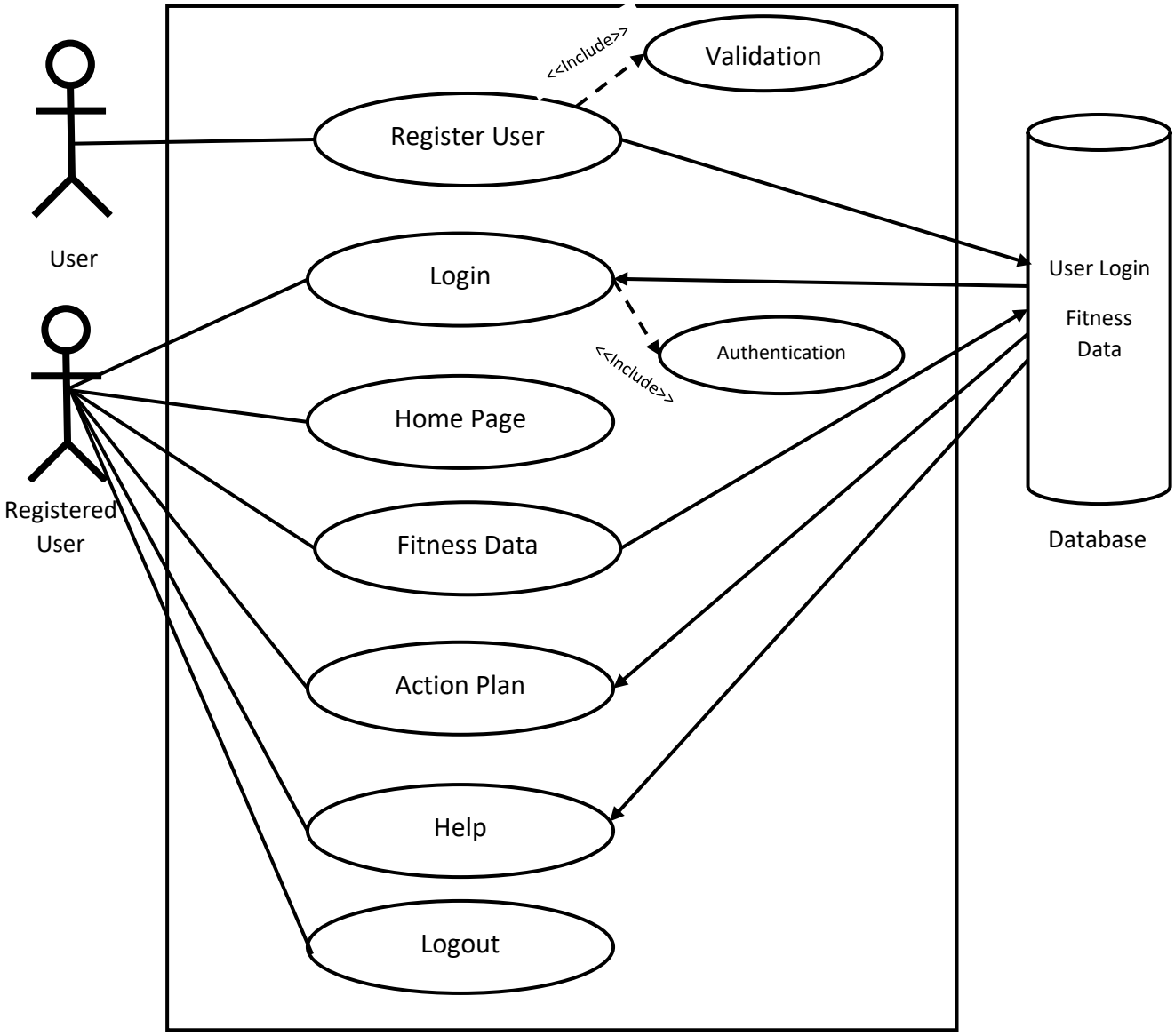
- My full completed project must be completed by the deadline of the 4th of March 2022.
- I ensure that I do not run into any copyright issues when it comes to the name of my project “YourFitnessFriend” and must to comply with the Copyright, Design and Patents Act 1988.
- I must ensure that there will be no costs involved during the course of my project as all of the software and materials are supplied by the school.
- I will be programming in Visual Basic within the Visual Studio 2010 software as I have several years of experience using this and I will be using EasyPHP as my Database Server while coding in MySQL as I believe it will be most suitable for my project.

Boundaries

The boundaries of my program will include:

- A login system which allow the end user to register a new username and password of their choice.
    - The username and password must contain more than 8 characters.
- The login system will use input validation, which will verify the credentials that the user has entered and allow them to be directed to the home page.
- When entering data users will not be able to:
    - Enter an Age less than 16, Weight less than 45kg or an Height less than 120cm
- The data will be calculated and stored within a Database to allow the fields to be stored externally rather than an array so it can be reused without re-entering.
- An insertion sort will be used to sort the fitness data, which will be displayed to the user.

UML: Use Case Diagram



Requirements Specification

For my program to be able to satisfy the requirements, it must meet several Criteria:

End User Requirements:

- The users for my system will be able to register a unique username and password of their choice.
- The user will be allowed to login using their own chosen username and password.
- The user will be able to easily navigate throughout my program.
- The user will be able to input their own fitness data into their program and be able to see their action plan.
- The user will also be able to access a help screen for tips to reach their desired goal.

Functional Requirements:

- My program will be able to store user credentials in an external database.
- My program will be able to validate all user inputs to make sure they are fit for purpose.
- My program will store the users data in an external database
- My program will display the user's action plan and a help screen.

Project Plan

Task	Sub Task	Duration	Start Date	Target Date
Analysis: **4 Hours And 30 Minutes** **(Over the course of 4 Days)**	Read AH Instructions for candidates and then generate my project idea	1 Hour	26th November	26th November
	Description of the problem: Outline of problem, Scope, Boundaries and Constraints	1 Hour 30 Minutes	3rd December	3rd December
	Create UML Use Case Diagram	45 Minutes	10th December	10th December
	Create Requirements Specifications for End User and Functional Requirements	45 Minutes	10th December	10th December
	Project Plan	30 Minutes	17th December	17th December
Design: **11 Hours And 15 Minutes** **(Over the course of 11 Days)**	Project Design - Pseudocode	4 Hour 30 Minutes	5th January	14th January
	User-interface Design – Wireframe	3 Hours	18th January	25st January
	Class Diagram	45 Minutes	26th January	26th January
	Design Data Dictionary	45 Minutes	28th January	28th January
	Design SQL	45 Minutes	28th January	28th January
Implementation: **8 Hours And 15 Minutes** **(Over the course of 9 Days)**	Create User Interface	3 Hours	1st February	8th February
	Create VB Program	3 Hour 45 Minutes	9th February	18th February
	Create Database	45 Minutes	9th February	9th February
	Write Description of new skills	45 Minutes	18th February	18th February
	Log of all ongoing testing	*Ongoing through implementation	1st February	18th February

Testing: **3 Hours And 45 Minutes** **(Over the course of 3 Days)**	Test Plan including Test Cases and Persona	1 Hour 30 Minutes	22th February	23rd February
	Implement Testing	1 Hour 30 Minutes	23rd February	25th February
	Report Testing Issues	45 Minutes	25th February	25th February
Evaluation: **1 Hours And 30 Minutes** **(Over the course of 2 Days)**	Write Project Evaluation	1 Hour 30 Minutes	1st March	2nd March
Finalise Project: **45 Minutes** **(Over the course of 1 Day)**	Finalise analysis Finalise design Finalise implementation Finalise testing and evaluation	45 Minutes	4th March	4th March
	Submit Project	----------	4th March	4th March

List of Resources

Analysis	• Microsoft Word 2016 • Google Chrome
Design	• Microsoft Word 2016 • Google Chrome • https://wireframe.cc/
Implementation	• Microsoft Word 2016 • Google Chrome • Visual Basic • EasyPHP – MySQL • https://iconscout.com/ • https://www.tailorbrands.com/
Testing	• Microsoft Word 2016 • Visual Basic • EasyPHP - MySQL
Evaluation	• Microsoft Word 2016

Design

Pseudocode

```
Create FitnessData Class:

      Class FitnessData IS {INTEGER id, STRING Gender, INTEGER Weight, INTEGER
Height, INTEGER Age, STRING Exercise, STRING LoseOrGain, INTEGER BMR, INTEGER
ExerciseBMR, INTEGER FitnessGoal}

      METHODS

      CONSTRUCTOR (INTEGER id, STRING Gender, INTEGER Weight, INTEGER Height,
INTEGER Age, STRING Exercise, STRING LoseOrGain, INTEGER BMR, INTEGER ExerciseBMR,
INTEGER FitnessGoal)

            DECLARE THIS.id INITIALLY id

            DECLARE THIS.Gender INITIALLY Gender

            DECLARE THIS.Weight INITIALLY Weight

            DECLARE THIS.Height INITIALLY Height

            DECLARE THIS.Age INITIALLY Age

            DECLARE THIS.Exercise INITIALLY Exercise

            DECLARE THIS.LoseOrGain INITIALLY LoseOrGain

            DECLARE THIS.BMR INITIALLY BMR

            DECLARE THIS.ExerciseBMR INITIALLY ExerciseBMR

            DECLARE THIS.FitnessGoal INITIALLY FitnessGoal

      END CONSTRUCTOR


      PROCEDURE UpdateWeight(INTEGER newWeight)

      SET THIS.Weight To newWeight

      END PROCEDURE


      PROCEDURE UpdateAge(INTEGER newAge)

      SET THIS.Age To newAge

      END PROCEDURE


      PROCEDURE UpdateExercise(INTEGER newExercise)

      SET THIS.Exercise To newExercise

      END PROCEDURE
```

```
FUNCTION GetId() RETURNS id
	RETURN THIS.id
	END FUNCTION

FUNCTION GetGender() RETURNS STRING
	RETURN THIS.Gender
	END FUNCTION

	FUNCTION GetWeight() RETURNS STRING
	RETURN THIS.Weight
	END FUNCTION

	FUNCTION GetHeight() RETURNS STRING
	RETURN THIS.Height
	END FUNCTION

	FUNCTION GetAge() RETURNS STRING
	RETURN THIS.Age
	END FUNCTION

	FUNCTION GetExercise() RETURNS STRING
	RETURN THIS.Exercise
	END FUNCTION

	FUNCTION GetLoseOrGain() RETURNS STRING
	RETURN THIS.LoseOrGain
	END FUNCTION

	FUNCTION GetBMR() RETURNS STRING
	RETURN THIS.BMR
	END FUNCTION
```

```
        FUNCTION GetExerciseBMR() RETURNS STRING
        RETURN THIS.ExerciseBMR
        END FUNCTION

        FUNCTION GetFitnessGoal() RETURNS STRING
        RETURN THIS.FitnessGoal
        END FUNCTION
        END CLASS
```

Register Account:

1.1 Create Connection String

1.2 Open Database Connection with Connection String

1.3 GET Username AND Password from USER

1.4 IF Username >= 8 AND Password >= 8 THEN

1.5 SEND Username AND Password TO DATABASE

1.6 Run Insert Query into Database

1.7 DISPLAY Register Successful

1.8 ELSEIF Username <> "" AND Password <> "" THEN

1.9 DISPLAY Register Unsuccessful As characters where not greater than 8

1.10 ELSEIF Username = "" AND Password = "" THEN

1.11 DISPLAY Register Unsuccessful as no characters where entered

1.12 ELSE DISPLAY Register Unsuccessful

1.13 END IF

1.14 Close Database Connection

Login:

2.1 Create Connection String

2.2 Open Database Connection

2.3 Load Records from Database

2.4 Run Select Query to Select Usernames and Passwords

2.5 GET Username and Password from USER

2.6 If Username = Username in Database AND Password = Password in Database THEN

2.7 DISPLAY Login Successful

2.8 ELSEIF Username <> Username in Database AND Password <> Password in Database THEN

2.9 DISPLAY Login Unsuccessful as no characters where entered

2.10 ELSE Login Unsuccessful

2.11 END IF

2.12 Close Database Connection

Calculate Fitness Data:

3.1 Create Connection String

3.2 Open Database Connection with Connection String

3.3 Run Query that Checks Username

3.4 GET Username from USER

3.5 GET Gender from USER

3.6 GET Weight from USER

3.7 GET Height from USER

3.8 GET Age from USER

3.9 GET Exercise from USER

3.10 GET LoseOrGain from USER

3.11 Run Query that Checks Username

3.12 Close Connection

3.13 If Weight < 45 THEN

3.14 Display Error

3.15 If Height < 120 THEN

3.16 Display Error

3.17 If Age < 16 THEN

3.18 Display Error

```
3.19 IF Gender = "Male" THEN
3.20 BMR = 66 + (13.7 * Weight) + (5 * Height) – (6.8 * Age)
3.21 ELSEIF Gender = "Female" THEN
3.22 BMR = 655 + (9.6 * Weight) + (1.8 * Height) – (4.7 * Age)
3.23 END IF
3.24 IF Exercise = "Sedentary" THEN
3.25 ExerciseBMR = BMR * 1.2
3.26 ELSEIF Exercise = "Lightly" THEN
3.27 ExerciseBMR = BMR * 1.375
3.28 ELSEIF Exercise = "Moderately" THEN
3.29 ExerciseBMR = BMR * 1.55
3.30 ELSEIF Exercise = "VeryActive" THEN
3.31 ExerciseBMR = BMR * 1.725
3.32 ELSEIF Exercise = "ExtraActive" THEN
3.33 ExerciseBMR = BMR * 1.9
3.34 END IF
3.35 IF LoseOrGain = "Lose" THEN
3.36 FitnessGoal = ExerciseBMR - 300
3.37 ELSEIF LoseOrGain = "Gain" THEN
3.38 FitnessGoal = ExerciseBMR + 300
3.39 END IF
3.40 Open Connection with Connection String
3.41 Run Insert Query that Inserts Fitness Data
3.42 Displays UserID
3.43 Close Database Connection
```

```
Show Fitness Data / Action Plan:
       4.1 RECORD Fitness
       4.2 Public id As Integer
       4.3 Public Gender As String
       4.4 Public Weight As Integer
       4.5 Public Height As Integer
       4.6 Public Age As Integer
       4.7 Public Exercise As String
       4.8 Public LoseOrGain As String
       4.9 Public BMR As Integer
       4.10 Public ExerciseBMR As Integer
       4.11 Public FitnessGoal As Integer
       4.12 End RECORD
       4.1 Create Connection String
       4.1 Open Database Connection with Connection String
       4.2 Run Query that selects UserID
       4.3 Display FitnessGoal
       4.4 Call FitnessData()
       4.5 Display Action Steps
       4.6 Generate Random Number between 1 and 5
       4.7 Run IF To determine meal plan
       4.8 Display Meal Plan
       4.9 Sub FitnessData()
       4.10 Dim fitness(9) As Fitness
      4.11 DECLARE tempFitness As Fitness
      4.12 DECLARE currentPosition As Integer
      4.13 DECLARE newPosition As Integer
      4.14 For currentPosition = 1 To 9
      4.15 tempFitness = fitness(currentPosition)
      4.16 newPosition = currentPosition
      4.17 While newPosition > 0 AndAlso fitness(newPosition - 1).FitnessGoal >
tempFitness.FitnessGoal
      4.18 fitness(newPosition) = fitness(newPosition - 1)
      4.19 newPosition = newPosition - 1
```

4.20 End While

4.21 fitness(newPosition) = tempFitness

4.22 Next

4.23 SEND FitnessData TO DISPLAY

Show Help:

5.1 Create Connection String

5.2 Open Database Connection with Connection String

5.3 Run Query that selects UserID

5.4 Display LoseOrGain

5.4 Display FitnessGoal

5.6 DISPLAY How to Lose / Gain weight

5.7 DISPLAY How to Gain Muscle

5.8 DISPLAY Gym work outs

5.9 DISPLAY Cardio Workouts

5.10 Close Database Connection

UML Class Diagram

FitnessData
+ id: Integer + Gender: String + Weight: Integer + Height: Integer + Age: Integer + Exercise: String + LoseOrGain: String + BMR: Integer + ExerciseBMR: Integer + FitnessGoal: Integer
+ UpdateWeight() + UpdateAge() + UpdateExercise() +GetId() +GetGender() +GetWeight() +GetHeight() +GetAge() +GetExercise() +GetLoseOrGain() +GetBMR() +GetExerciseBMR() +GetFitnessGoal()

Data Dictionary (Database Integration)

User Login Table:

Entity: User Login					
Attribute Name	Key	Type	Size	Required	Validation
UsernameData		Varchar	20	Y	
PasswordData		Varchar	20	Y	
UserID	PK	Integer		Y	

Sample Data: User Login		
UserID	UsernameData	PasswordData
1	Zayd3030	LetMeIn54!
2	BilartCoin45	Password123

Fitness Data Table:

Entity: Fitness Data					
Attribute Name	Key	Type	Size	Required	Validation
id	PK	Integer		Y	AUTO_INCREMENT
Gender		Varchar	6	Y	Restricted Choice Male or Female
Weight		Integer		Y	Range Check Weight >= 45
Height		Integer		Y	Range Check Height >= 120
Age		Integer		Y	Range Check Age >= 16
Exercise		Varchar	20	Y	
LoseOrGain		Varchar	4	Y	
BMR		Integer		Y	
ExerciseBMR		Integer		Y	
FitnessGoal		Integer		Y	
UserID	FK	Integer		Y	

	Sample Data: Fitness Data									
id	UserID	Gender	Weight	Height	Age	Exercise	LoseOrGain	BMR	ExerciseBMR	FitnessGoal
1	1	Male	75	176	35	VeryActive	Lose	1736	2995	2695
2	2	Male	61	168	18	Moderately	Gain	2526	2526	2526

Query Design

Register User:

INSERT	UserLogin
VALUES	UsernameData, PasswordData

Login User:

SELECT	*
FROM	UserLogin
WHERE	UsernameData = EnterUsername.Text AND PasswordData = EnterPassword.Text <FROM LOGIN FORM>

Select User ID:

SELECT	UserId
FROM	UserLogin
WHERE	UsernameData = UsernameEnter.Text <FROM ENTER FITNESS DATA FORM>

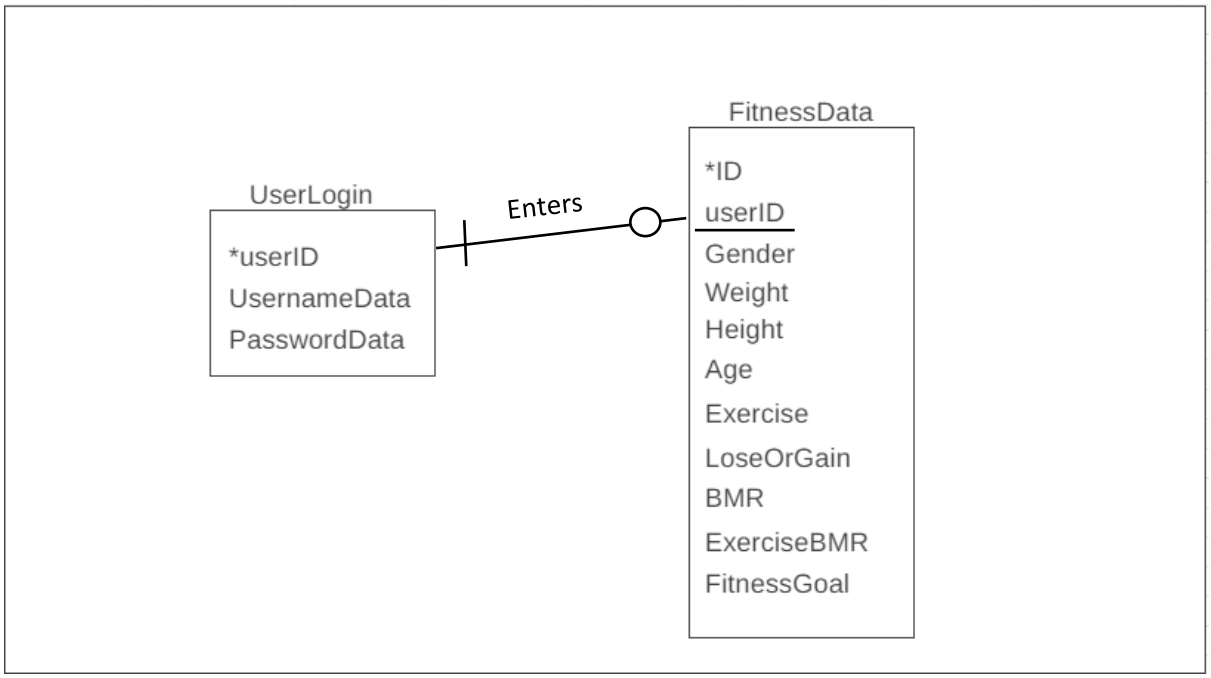
Insert Fitness Data:

INSERT	FitnessData
VALUES	UserId, Gender, Weight, Height, Age, Exercise, LoseOrGain, BMR, ExerciseBMR, FitnessGoal

Select Fitness Data For Action Plan:

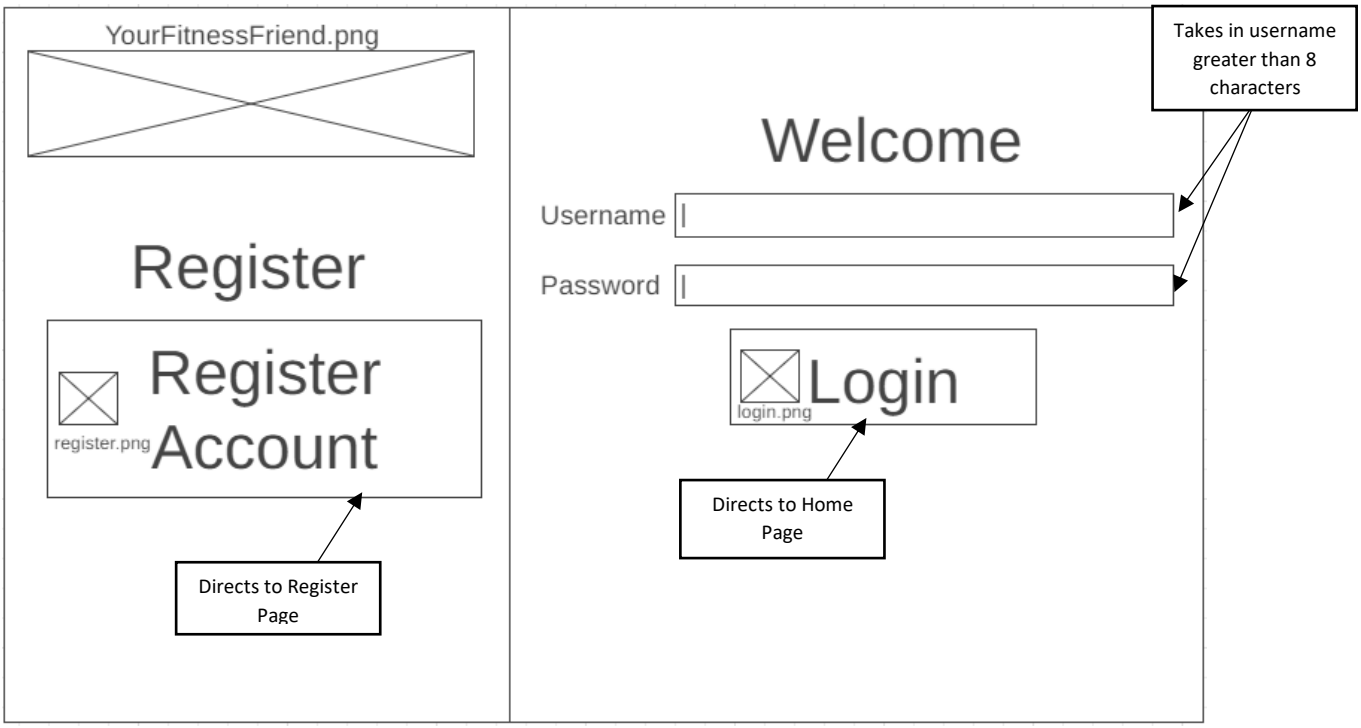
SELECT	*
FROM	FitnessData, UserLogin
WHERE	FitnessData.userId = UserLogin.userId AND FitnessData.userId = EnterID.Text <FROM ACTION PLAN FORM>

Entity Relationship Diagram

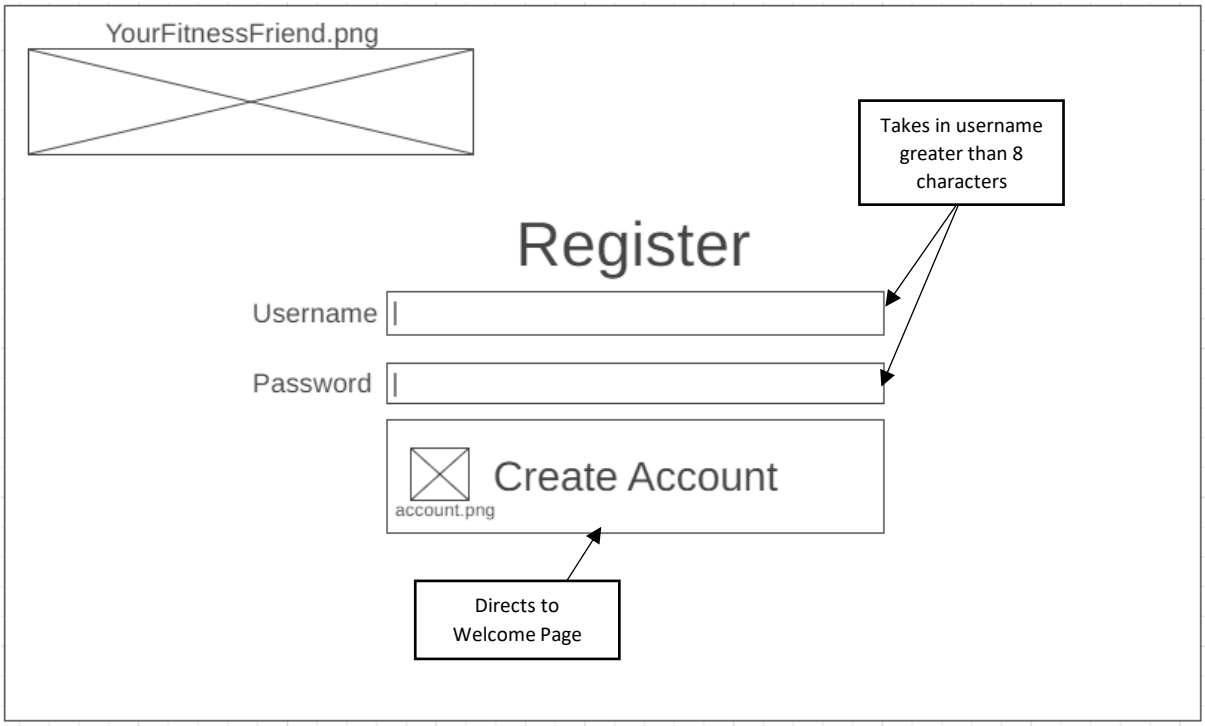


User Interface (Wireframes)

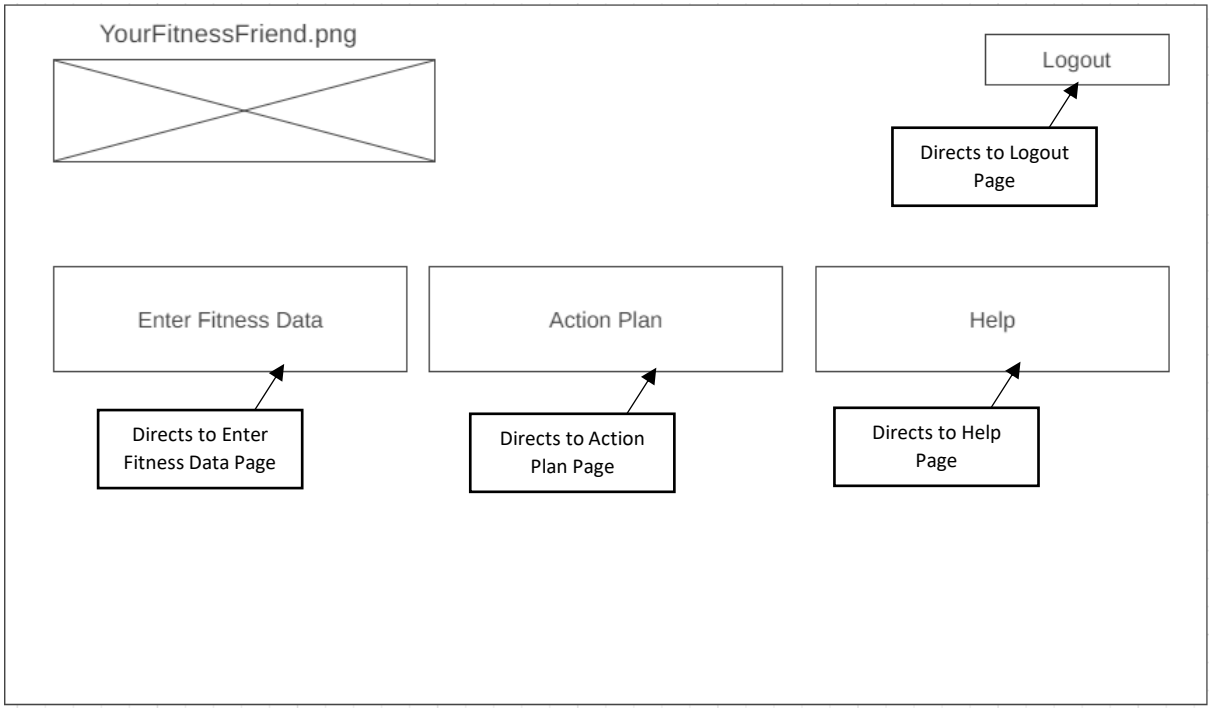
Welcome Page:



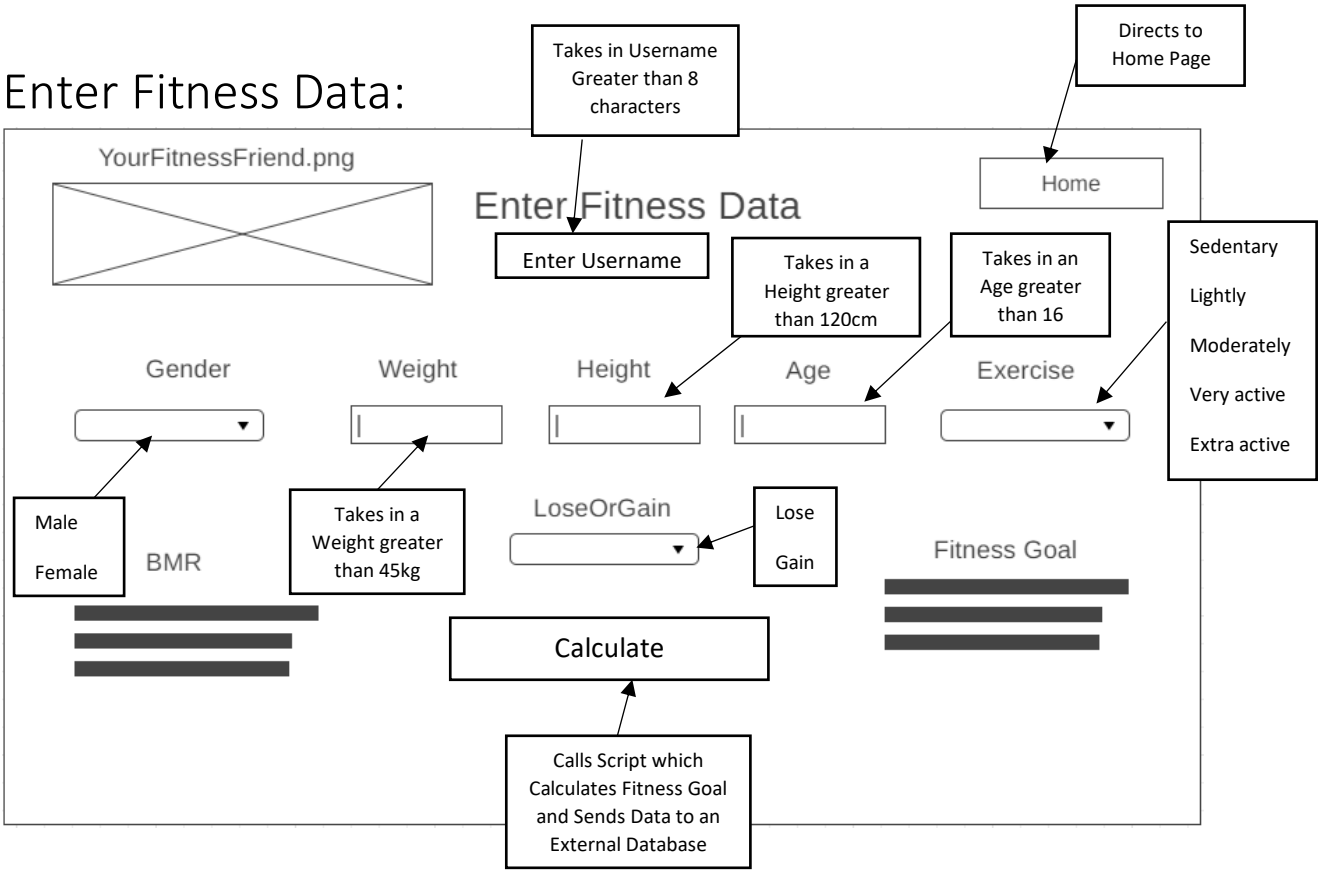
Register Page:



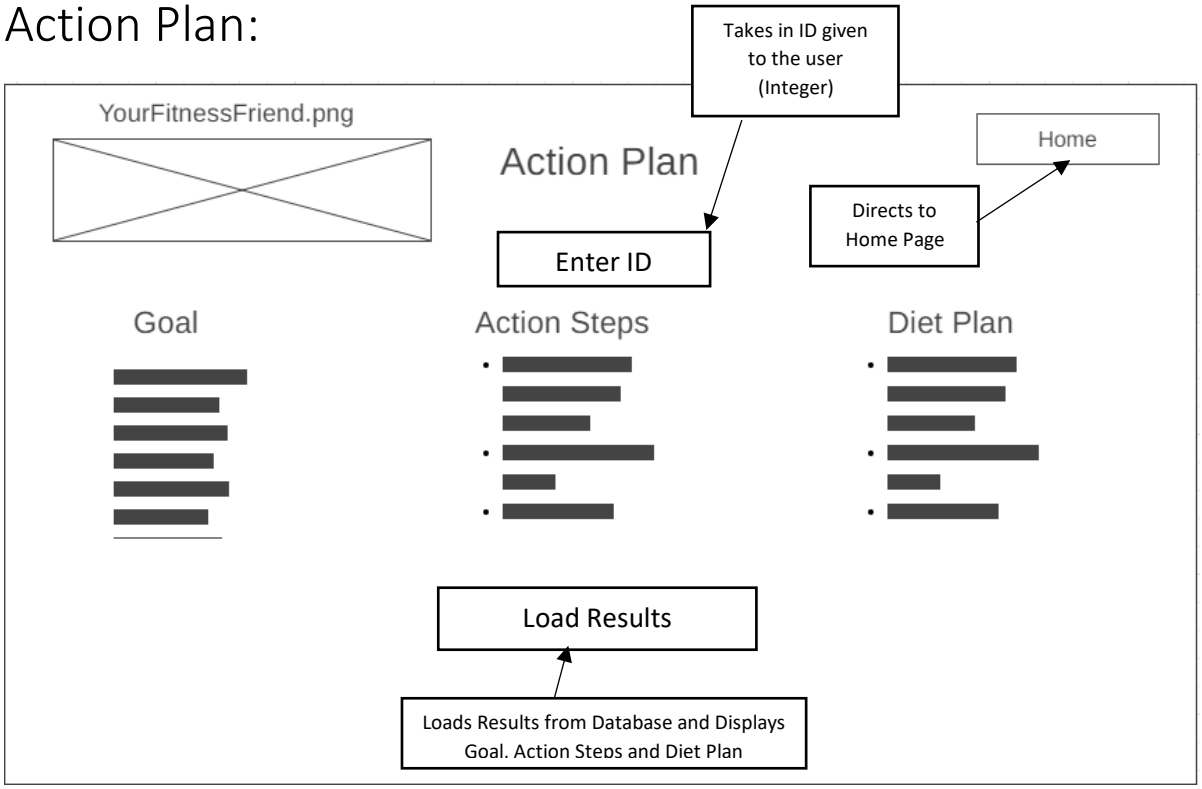
Home Page:



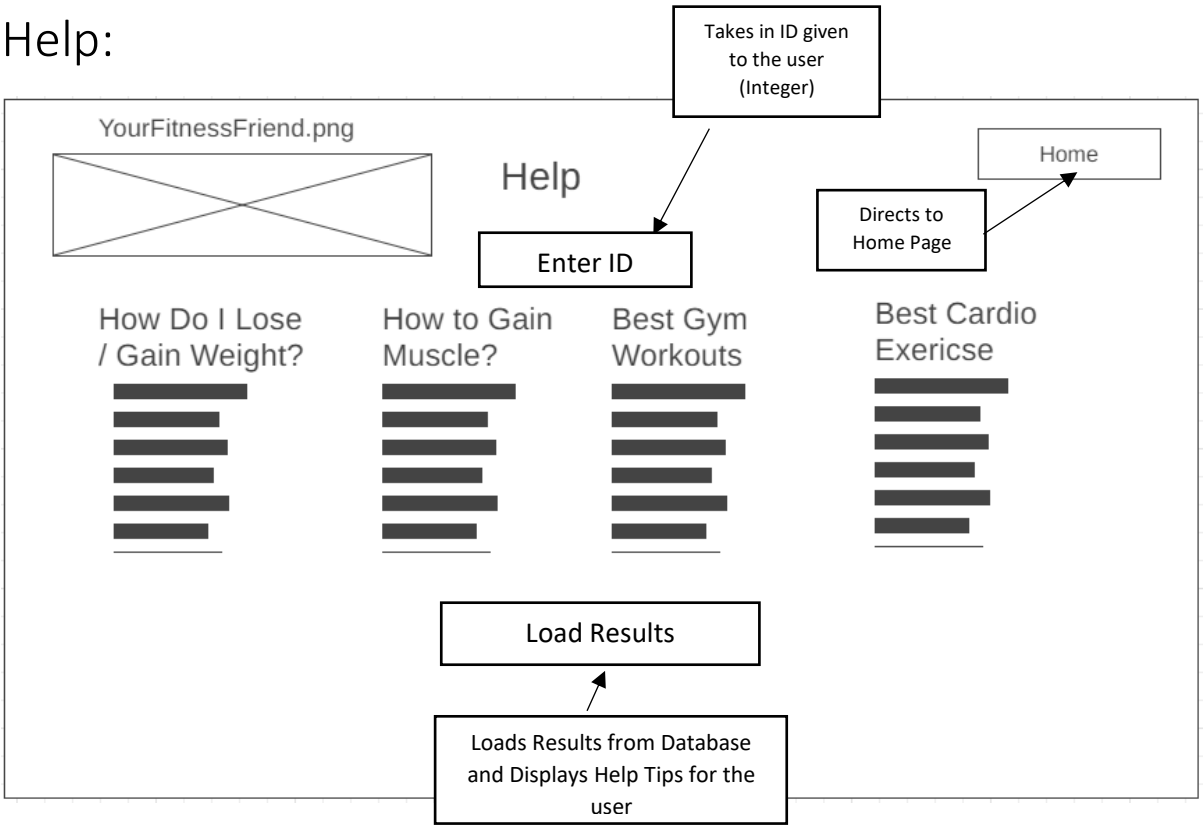
Enter Fitness Data:



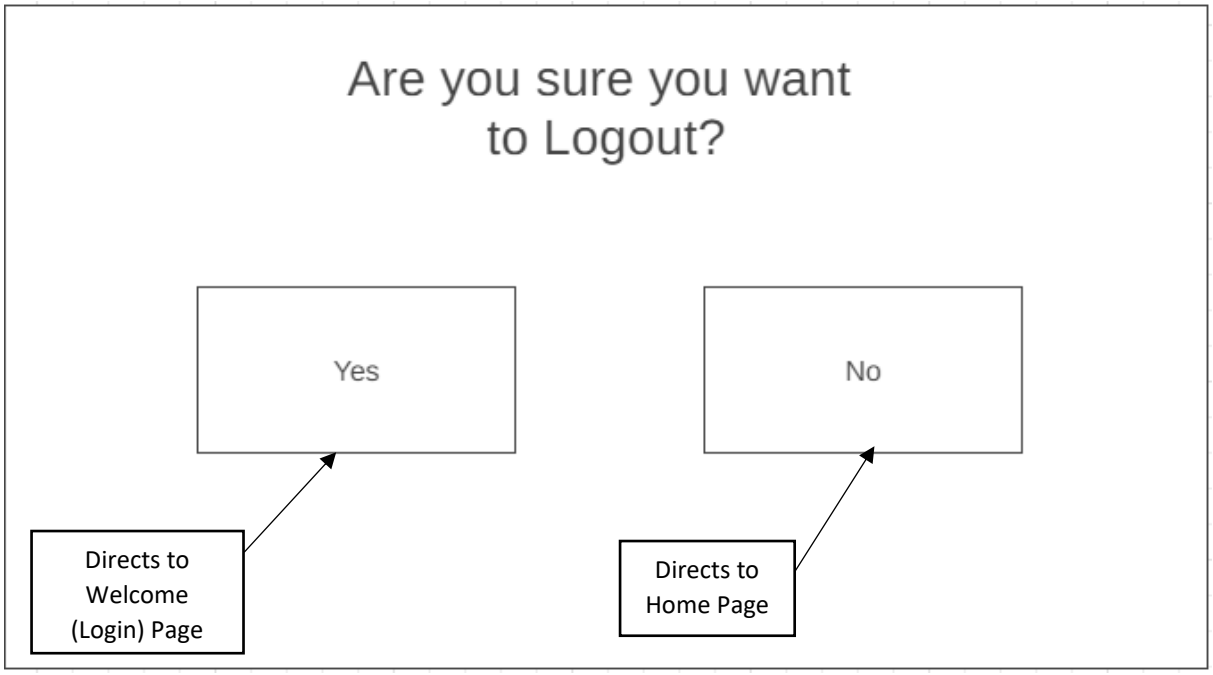
Action Plan:



Help:



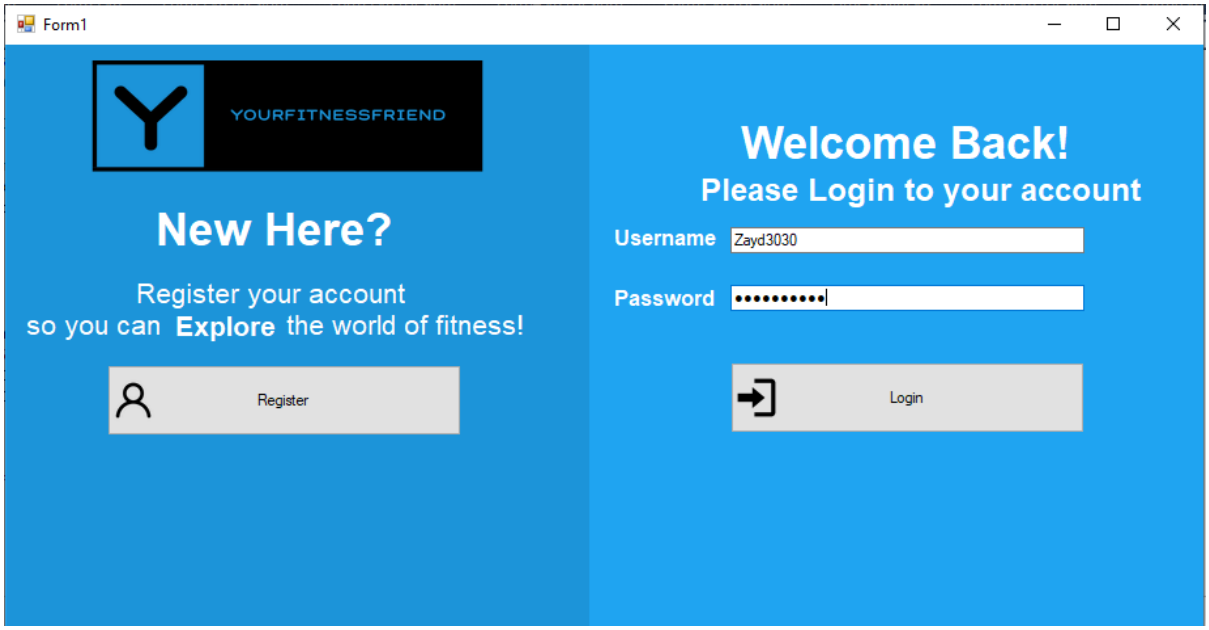
Logout:


Are you sure you want to Logout?
Yes
No
Directs to Welcome (Login) Page
Directs to Home Page

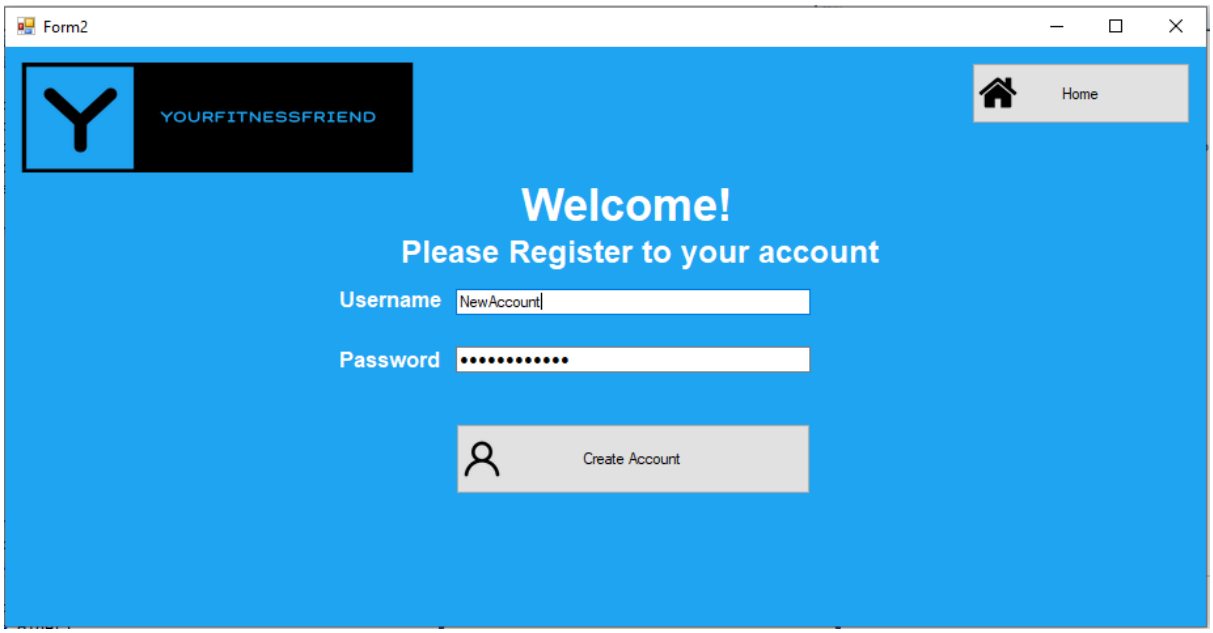
Implementation

Software Design and Development User Interface

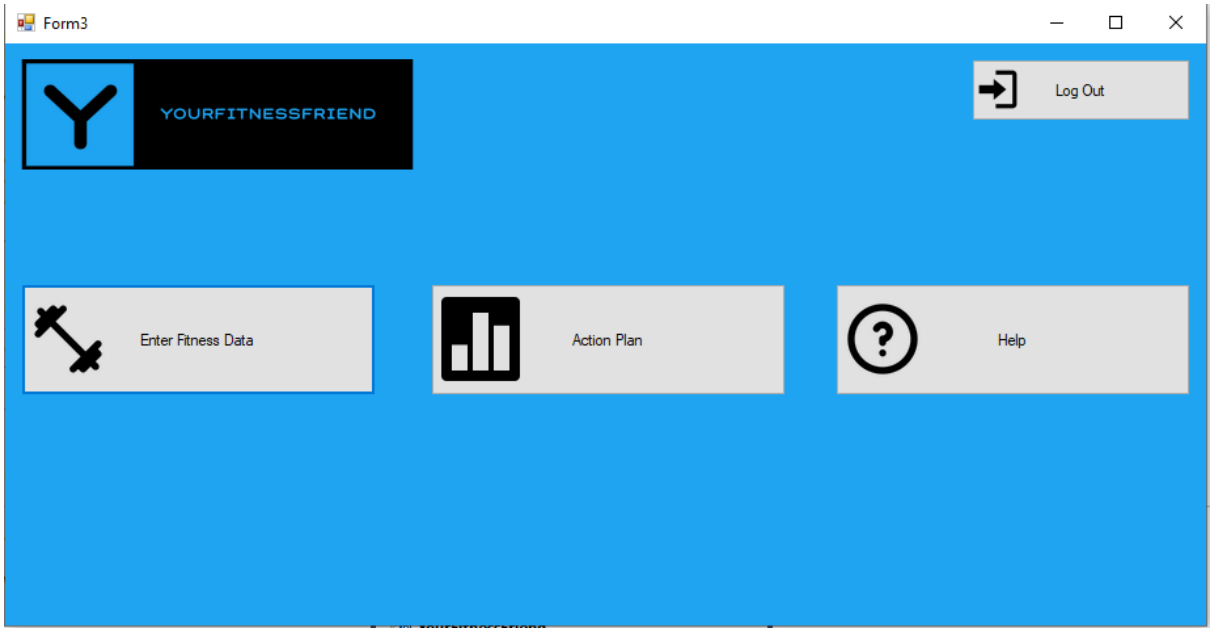
Welcome (Login) Page:



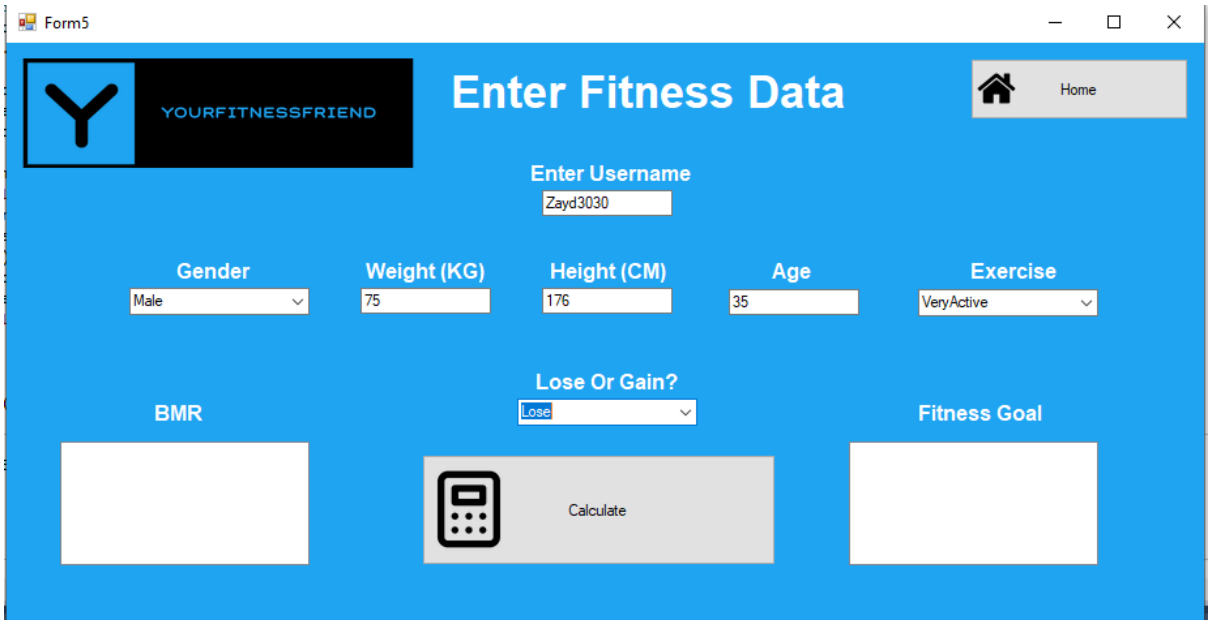
Register Page:



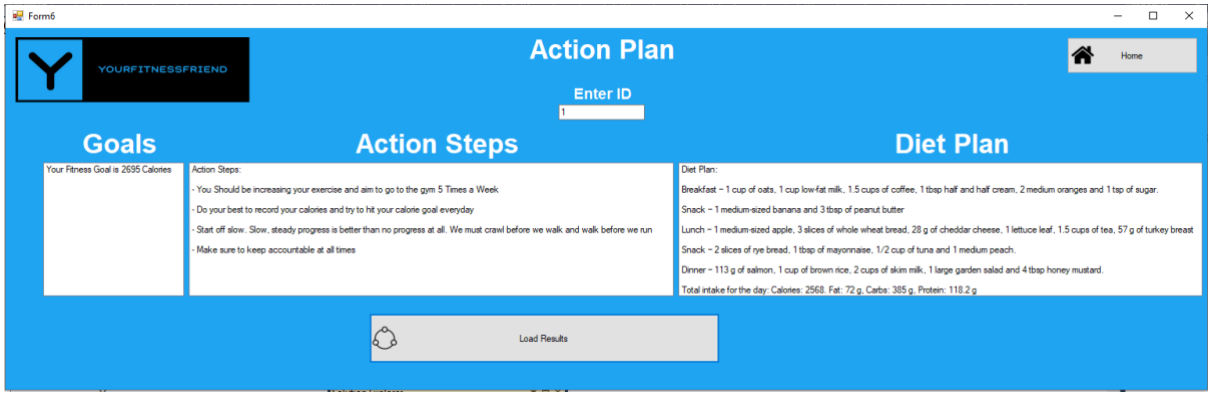
Home Page:



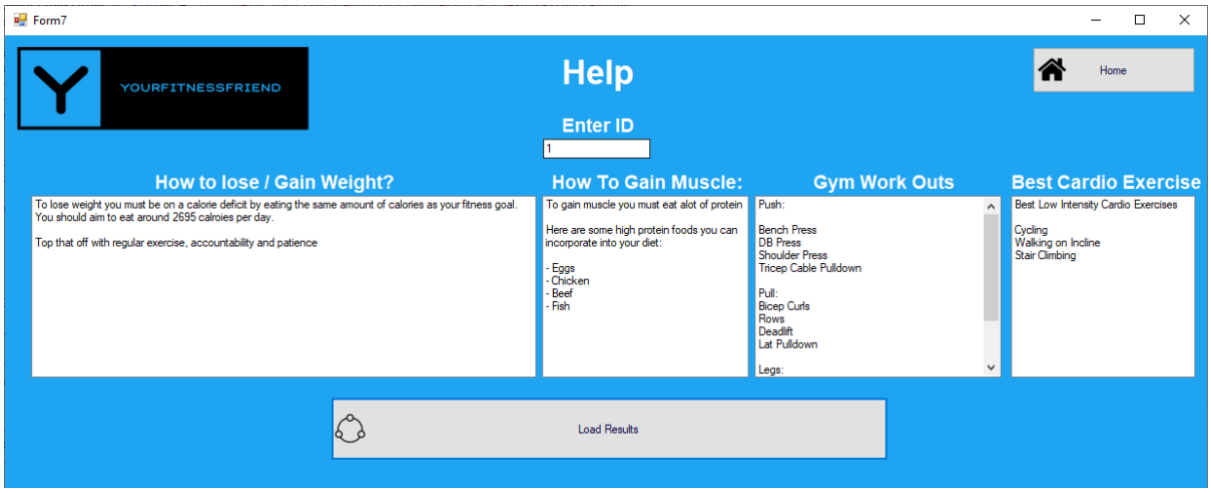
Enter Fitness Data Page:



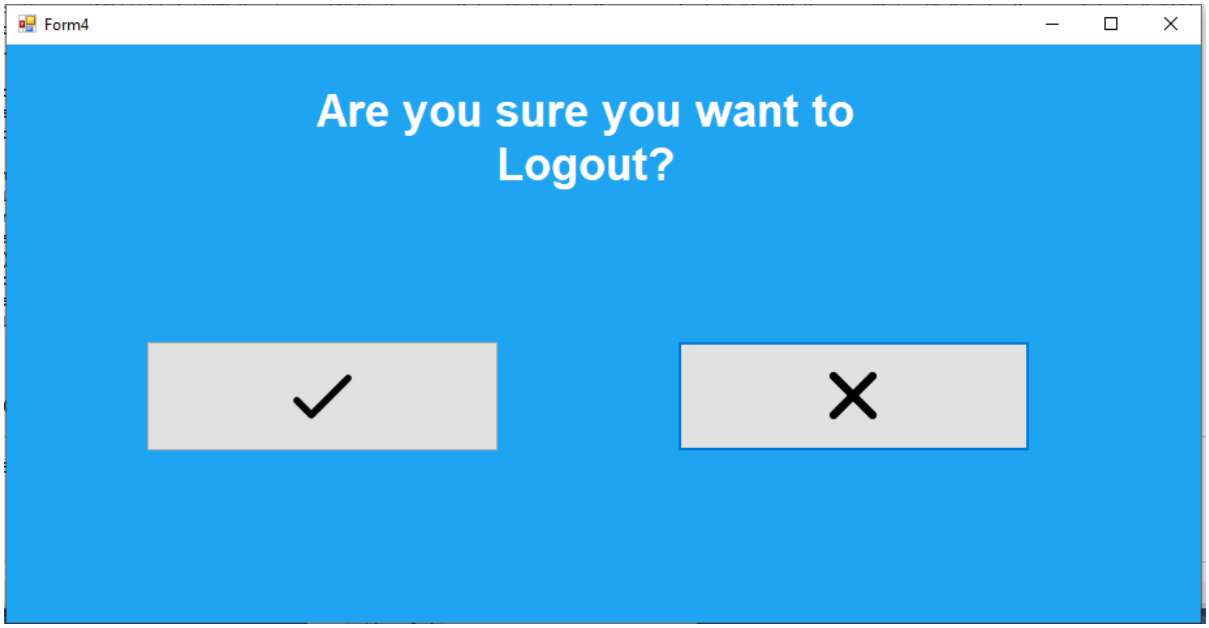
Action Plan Page:



Help Page:

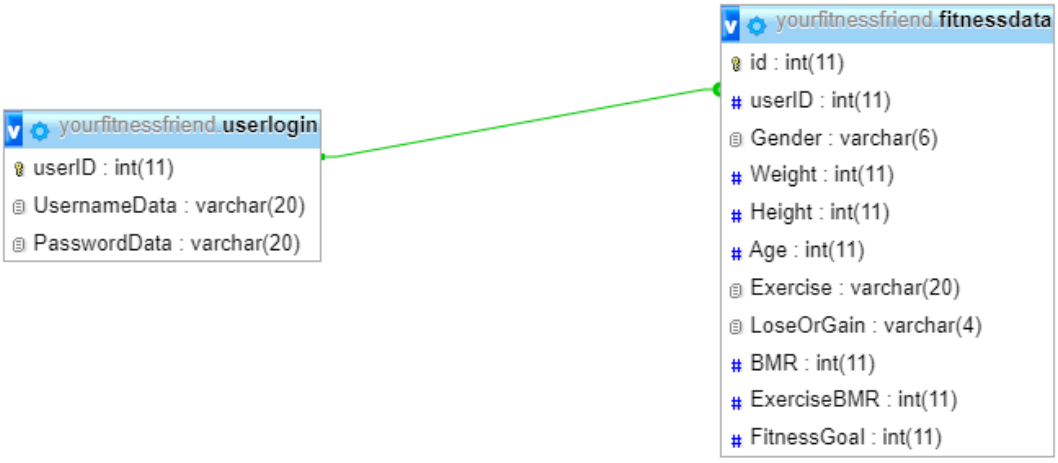


Logout Page:

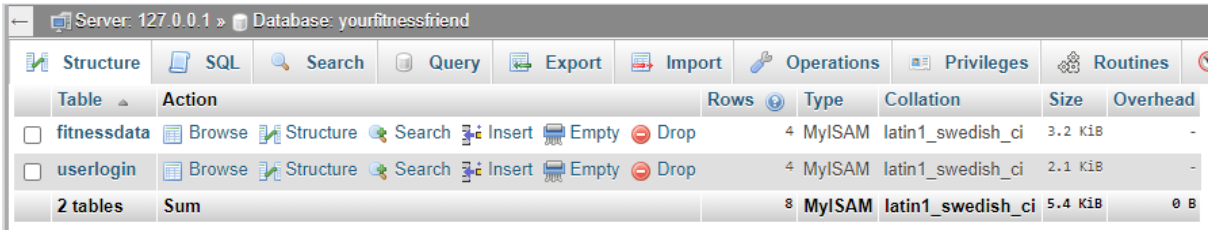


Database Design and Development Implemented Tables

Table Relationship:



Database Contents:



User Login Sample Data:

	userID	UsernameData	PasswordData
Edit Copy Delete	1	Zayd3030	LetMeIn54!
Edit Copy Delete	2	BilartCoin45	Password123
Edit Copy Delete	3	TheUsername	ThePassword
Edit Copy Delete	4	Admin12345	qwerty1234

Fitness Data Sample Data:

	id	userID	Gender	Weight	Height	Age	Exercise	LoseOrGain	BMR	ExerciseBMR	FitnessGoal
Edit Copy Delete	1	1	Male	75	176	35	VeryActive	Lose	1736	2995	2695
Edit Copy Delete	2	2	Male	61	168	18	Lightly	Gain	1619	2226	2526
Edit Copy Delete	3	3	Female	50	150	23	ExtraActive	Lose	1297	2464	2164
Edit Copy Delete	4	4	Male	45	160	18	ExtraActive	Gain	1360	2584	2884

Software Design and Development Program Code

FitnessData Class:

```
Public Class FitnessData
    Public id As Integer
    Public Gender As String
    Public Weight As Integer
    Public Height As Integer
    Public Age As Integer
    Public Exercise As String
    Public LoseOrGain As String
    Public BMR As Integer
    Public ExerciseBMR As Integer
    Public FitnessGoal As Integer

    Public Sub New(ByVal id As Integer, ByVal Gender As String, ByVal Weight As Integer,
ByVal Height As Integer, ByVal Age As Integer, ByVal Exercise As String, ByVal LoseOrGain
As String)
        Me.id = id
        Me.Gender = Gender
        Me.Weight = Weight
        Me.Height = Height
        Me.Age = Age
        Me.Exercise = Exercise
        Me.LoseOrGain = LoseOrGain
    End Sub
    Public Sub UpdateWeight(ByVal NewWeight As Integer)
        Me.Weight = NewWeight
    End Sub

    Public Sub UpdateAge(ByVal NewAge As Integer)
        Me.Age = NewAge
    End Sub

    Public Sub UpdateExercise(ByVal NewExercise As String)
        Me.Exercise = NewExercise
    End Sub
    Public Function GetId(ByVal id As Integer)
        Return id
    End Function

    Public Function GetGender(ByVal Gender As String)
        Return Gender
    End Function

    Public Function GetWeight(ByVal Weight As String)
        Return Weight
    End Function

    Public Function GetHeight(ByVal Height As String)
        Return Height
    End Function

    Public Function GetAge(ByVal Age As String)
        Return Age
    End Function

    Public Function GetExercise(ByVal Exercise As String)
        Return Exercise
    End Function
```

Welcome (Login) Page:

```
Imports MySql.Data.MySqlClient
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnLogin.Click
        Dim connection As MySqlConnection
        Dim command As MySqlCommand
        Dim connectionString As String =
"server=localhost;user=root;password=;database=YourFitnessFriend"
        Dim reader As MySqlDataReader

        connection = New MySqlConnection(connectionString)
        connection.Open()
        command = New MySqlCommand("SELECT * FROM UserLogin WHERE UsernameData='" &
EnterUsername.Text & "' AND PasswordData='" & EnterPassword.Text & "'")
        command.Connection = connection
        reader = command.ExecuteReader()

        While reader.Read()
            Dim GetUsername As String = reader.GetString(1).ToString
            Dim GetPassword As String = reader.GetString(2).ToString

            If GetUsername = EnterUsername.Text And GetPassword = EnterPassword.Text Then
                MsgBox("Login Successful")
                EnterUsername.Clear()
                EnterPassword.Clear()
                Me.Hide()
                Form3.Show()
            ElseIf GetUsername <> EnterUsername.Text Or GetPassword <> EnterPassword.Text
Then
                MsgBox("Login Failed")
            End If

        End While
        connection.Close()
    End Sub

    Private Sub RegisterBtn_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RegisterBtn.Click
        Me.Hide()
        Form2.Show()
    End Sub
End Class
```

Register Page:

```
Imports MySql.Data.MySqlClient
Public Class Form2
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnCreate.Click
        Dim connection As MySqlConnection
        Dim command As MySqlCommand
        Dim connectionString As String =
"server=localhost;user=root;password=;database=YourFitnessFriend"

        connection = New MySqlConnection(connectionString)
        Dim Username = EnterUsername.Text
        Dim Password = EnterPassword.Text

        connection = New MySqlConnection(connectionString)
        command = New MySqlCommand("INSERT INTO UserLogin (UsernameData, PasswordData)
VALUES('" & Username & "','" & Password & "')")
        command.Connection = connection

        connection.Open()
        If Len(Username) >= 8 Or Len(Password) >= 8 Then
            command.ExecuteNonQuery()
            MsgBox("Account Created")
            Me.Hide()
            Form1.Show()
        ElseIf Username <> "" Or Password <> "" Then
            MsgBox("Error - Please Enter a Username and Password Greater Than 8
Characters")
        ElseIf Username = "" Or Password = "" Then
            MsgBox("Error - Please Enter a Username and Password")
        End If
        connection.Close()
    End Sub

    Private Sub BtnHome_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnHome.Click
        Me.Hide()
        Form1.Show()
    End Sub
End Class
```

Home Page:

```
Public Class Form3

    Private Sub BtnLogOut_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnLogOut.Click
        Me.Hide()
        Form4.Show()
    End Sub

    Private Sub BtnFitnessData_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnFitnessData.Click
        Me.Hide()
        Form5.Show()
    End Sub

    Private Sub BtnAction_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnAction.Click
        Me.Hide()
        Form6.Show()
    End Sub

    Private Sub BtnHelp_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnHelp.Click
        Me.Hide()
        Form7.Show()
    End Sub
End Class
```

Enter Fitness Data Page:

```
Imports MySql.Data.MySqlClient
Public Class Form5
    Private Sub BtnFitnessData_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnFitnessData.Click
        Dim Person1 As New FitnessData(0, ComboBoxGender.Text, TextBoxWeight.Text,
TextBoxHeight.Text, TextBoxAge.Text, ComboBoxExercise.Text, ComboBoxLoseOrGain.Text)
        If Person1.Gender = "Male" Then
            Person1.BMR = 66 + (13.7 * Person1.Weight) + (5 * Person1.Height) - (6.8 *
Person1.Age)
        ElseIf Person1.Gender = "Female" Then
            Person1.BMR = 655 + (9.6 * Person1.Weight) + (1.8 * Person1.Height) - (4.7
* Person1.Age)
        End If

        If Person1.Exercise = "Sedentary" Then
            Person1.ExerciseBMR = Person1.BMR * 1.2
        ElseIf Person1.Exercise = "Lightly" Then
            Person1.ExerciseBMR = Person1.BMR * 1.375
        ElseIf Person1.Exercise = "Moderately" Then
            Person1.ExerciseBMR = Person1.BMR * 1.55
        ElseIf Person1.Exercise = "VeryActive" Then
            Person1.ExerciseBMR = Person1.BMR * 1.725
        ElseIf Person1.Exercise = "ExtraActive" Then
            Person1.ExerciseBMR = Person1.BMR * 1.9
        End If

        If Person1.LoseOrGain = "Lose" Then
            Person1.FitnessGoal = Person1.ExerciseBMR - 300
        ElseIf Person1.LoseOrGain = "Gain" Then
            Person1.FitnessGoal = Person1.ExerciseBMR + 300
        End If


        LstDisplayBMR.Items.Add("Your Natural BMR is = " & Person1.BMR)
        LstDisplayBMR.Items.Add("Your BMR is = " & Person1.ExerciseBMR)
        LstDisplayFG.Items.Add("Your Fitness Goal is = " & Person1.FitnessGoal)

        Call SendData(Person1)
    End Sub

    Private Sub BtnHome_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnHome.Click
        EnterUsername.Clear()
        TextBoxHeight.Clear()
        TextBoxWeight.Clear()
        TextBoxAge.Clear()
        ComboBoxExercise.SelectedIndex = -1
        ComboBoxGender.SelectedIndex = -1
        ComboBoxLoseOrGain.SelectedIndex = -1
        LstDisplayBMR.Items.Clear()
        LstDisplayFG.Items.Clear()
        Me.Hide()
        Form3.Show()
    End Sub
    Sub SendData(ByVal Person1 As FitnessData)

        Dim id = Person1.id
        Dim Gender = Person1.Gender
        Dim Weight = Person1.Weight
        Dim Height = Person1.Height
```

```
        Dim Age = Person1.Age
        Dim Exercise = Person1.Exercise
        Dim LoseOrGain = Person1.LoseOrGain
        Dim BMR = Person1.BMR
        Dim ExerciseBMR = Person1.ExerciseBMR
        Dim FitnessGoal = Person1.FitnessGoal
        Dim Newid As Integer
        Dim UsernameEnter As String = EnterUsername.Text

        Dim connection As MySqlConnection
        Dim command As MySqlCommand
        Dim connectionString As String =
"server=localhost;user=root;password=;database=YourFitnessFriend"
        Dim reader As MySqlDataReader

        connection = New MySqlConnection(connectionString)

        connection.Open()
        command = New MySqlCommand("SELECT UserId FROM UserLogin WHERE UsernameData
='" & UsernameEnter & "'")
        command.Connection = connection
        reader = command.ExecuteReader

        While reader.Read()
            Newid = reader.Item("UserId")
        End While
        connection.Close()


        connection.Open()
        command = New MySqlCommand("INSERT INTO FitnessData(UserId, Gender, Weight,
Height, Age, Exercise, LoseOrGain, BMR, ExerciseBMR, FitnessGoal) VALUES('" & Newid &
"','" & Gender & "','" & Weight & "','" & Height & "','" & Age & "','" & Exercise &
"','" & LoseOrGain & "','" & BMR & "','" & ExerciseBMR & "','" & FitnessGoal & "')")
        command.Connection = connection
        If Person1.Weight < 45 Then
            MsgBox("Error - Please Enter a Weight Over 45 KG")
        ElseIf Person1.Height < 120 Then
            MsgBox("Error - Please Enter a Height Over 120 CM")
        ElseIf Person1.Age < 16 Then
            MsgBox("Error - Please Enter an Age Over 16")
        Else
            command.ExecuteNonQuery()
            MsgBox("Data Added")
            MsgBox("Your ID is " & Newid)

            EnterUsername.Clear()
            TextBoxHeight.Clear()
            TextBoxWeight.Clear()
            TextBoxAge.Clear()
            ComboBoxExercise.SelectedIndex = -1
            ComboBoxGender.SelectedIndex = -1
            ComboBoxLoseOrGain.SelectedIndex = -1
            LstDisplayBMR.Items.Clear()
            LstDisplayFG.Items.Clear()

        End If

        connection.Close()
    End Sub
End Class
```

Action Plan Page:

```
Imports MySql.Data.MySqlClient
Public Structure Fitness
    Public id As Integer
    Public Gender As String
    Public Weight As Integer
    Public Height As Integer
    Public Age As Integer
    Public Exercise As String
    Public LoseOrGain As String
    Public BMR As Integer
    Public ExerciseBMR As Integer
    Public FitnessGoal As Integer
End Structure
Public Class Form6
    Private Sub BtnHome_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnHome.Click
        LstDisplayGoal.Items.Clear()
        LstDisplayAS.Items.Clear()
        LstDisplayDP.Items.Clear()
        EnterID.Clear()
        Me.Hide()
        Form3.Show()
    End Sub

    Private Sub BtnLoad_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnLoad.Click
        Dim fitness As Fitness
        Dim connection As MySqlConnection
        Dim command As MySqlCommand
        Dim connectionString As String =
"server=localhost;user=root;password=;database=YourFitnessFriend"
        Dim reader As MySqlDataReader
        Dim myRnd As Integer
        Dim days As Integer

        connection = New MySqlConnection(connectionString)
        connection.Open()

        command = New MySqlCommand("SELECT * FROM FitnessData, UserLogin WHERE
FitnessData.userId = UserLogin.userId AND FitnessData.userId = '" & EnterID.Text &
"'")
        command.Connection = connection
        reader = command.ExecuteReader

        While reader.Read()
            fitness.FitnessGoal = reader.Item("FitnessGoal")
            Call InsertionSort()
            LstDisplayGoal.Items.Add("Your Fitness Goal is " & fitness.FitnessGoal & "
Calories ")

            fitness.Exercise = reader.Item("Exercise")

            If fitness.Exercise = "Sedentary" Then
                days = 3
            ElseIf fitness.Exercise = "Lightly" Then
                days = 3
            ElseIf fitness.Exercise = "Moderately" Then
                days = 4
            ElseIf fitness.Exercise = "VeryActive" Then
                days = 5
```

```
            ElseIf fitness.Exercise = "ExtraActive" Then
                days = 6
            End If

            LstDisplayAS.Items.Add("Action Steps: ")
            LstDisplayAS.Items.Add("")
            LstDisplayAS.Items.Add("- You Should be increasing your exercise and aim
to go to the gym " & days & " Times a Week")
            LstDisplayAS.Items.Add("")
            LstDisplayAS.Items.Add("- Do your best to record your calories and try to
hit your calorie goal everyday")
            LstDisplayAS.Items.Add("")
            LstDisplayAS.Items.Add("- Start off slow. Slow, steady progress is better
than no progress at all. We must crawl before we walk and walk before we run")
            LstDisplayAS.Items.Add("")
            LstDisplayAS.Items.Add("- Make sure to keep accountable at all times")


            myRnd = CInt(Int((5 * Rnd()) + 1))

            If fitness.FitnessGoal > 2500 Then
                LstDisplayDP.Items.Add("Diet Plan: ")
                LstDisplayDP.Items.Add("")
                If myRnd = 1 Then
                    LstDisplayDP.Items.Add("Breakfast – One scoop of whey protein, 20
grams of blueberries, 2 whole eggs and 5 egg whites.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack – 2 small bananas and one scoop of
protein powder")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Lunch – 125 g of cooked brown rice, 198 g
of cooked chicken breast, 80 g of mixed salad (vegetables of your choice).")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Afternoon snack– 4 boiled egg whites, 1
apple and 2 scoops of whey protein powder (mixed with water).")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Dinner – 198 g of chicken breast, 60 g of
sweet potato and 80 g of mixed salad.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Total intake for the day: Calories: 2,347.
Fat: 32 g, Carbs: 159 g, Protein: 285.8 g")
                ElseIf myRnd = 2 Then
                    LstDisplayDP.Items.Add("Breakfast – 268 g of egg whites, two
slices of sprouted bread, reduced sugar jam and sugar-free ketchup.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack – Protein bar (68 g) and black
coffee.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Lunch – 74 g of oats, one medium-sized
banana, 10 g of coconut flakes and 3 pieces of 70% dark chocolate.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack – One large nectarine.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Lunch #2 – 150 g lean ground beef, 1 cup
of cauliflower florets, 150 g of Jasmine rice and 1 tbsp hoisin sauce.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Dinner – 1 cup of broccoli florets, 1 can
of tuna and one medium-size sweet potato (baked).")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Dessert – 2 low-calorie ice cream bars.")
                    LstDisplayDP.Items.Add("")
```

```
                    LstDisplayDP.Items.Add("Total intake for the day:  Calories:
2,413. Fat: 69.3 g, Carbs: 311.9 g, Protein: 193.7 g")
                ElseIf myRnd = 3 Then
                    LstDisplayDP.Items.Add("Breakfast – 2 6-inch buttermilk pancakes,
2 slices bacon, 2 pats butter, 3 tbsp pure maple syrup.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack – 1 medium banana and 1 cup
oatmeal.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Lunch – 1 bagel, 113 g sliced turkey, 2
slices tomato, 1 lettuce leaf, 1 slice cheddar cheese, 1 tsp mustard, 2 slices
avocado.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack – 1 container Greek yogurt, 1/2 cup
raspberries and 28 g pretzels.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Dinner – 113 g chicken breast, 1 cup white
rice, 1/2 chopped bell pepper, green onion, 1/2 red onion, 1/2 cup mushrooms, 2 tbsp
soy sauce and 1 egg")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Total intake for the day: Calories: 2448.
Fat: 79 g, Carbs: 301 g, Protein: 134 g")
                ElseIf myRnd = 4 Then
                    LstDisplayDP.Items.Add("Breakfast – 1 cup of oats, 1 cup low-fat
milk, 1.5 cups of coffee, 1 tbsp half and half cream, 2 medium oranges and 1 tsp of
sugar.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack – 1 medium-sized banana and 3 tbsp
of peanut butter")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Lunch – 1 medium-sized apple, 3 slices of
whole wheat bread, 28 g of cheddar cheese, 1 lettuce leaf, 1.5 cups of tea, 57 g of
turkey breast")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack – 2 slices of rye bread, 1 tbsp of
mayonnaise, 1/2 cup of tuna and 1 medium peach.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Dinner – 113 g of salmon, 1 cup of brown
rice, 2 cups of skim milk, 1 large garden salad and 4 tbsp honey mustard.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Total intake for the day: Calories: 2568.
Fat: 72 g, Carbs: 385 g, Protein: 118.2 g")
                ElseIf myRnd = 5 Then
                    LstDisplayDP.Items.Add("Breakfast – 1/2 cup of oats, 1 cup of
strawberries, 1 large egg and 3 egg whites (boiled), 1.25 cups of almond milk and 1
scoop of whey protein.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack – 227 g of shredded chicken breast,
2 low-carb whole-wheat wraps and 1 cup of bell peppers.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Lunch – 4 cups of romaine lettuce and
spinach mix, 12 grape tomatoes, 2 reduced-fat cheese sticks, 150 g Greek yogurt and 1
cup of roasted almonds.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack – 1.25 cups of unsweetened almond
milk, 1 scoop of whey protein, 150 g of greek yogurt and 1 cup of mixed frozen
fruits.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack – 1 scoop of whey protein mixed with
water.")
                    LstDisplayDP.Items.Add("")
```

```
                    LstDisplayDP.Items.Add("Dinner – 226 g of lean ground beef, 1/2
cup of brown rice and 2 cups of steamed broccoli.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Dessert: 1 cup of low-fat cottage cheese
and 1 cup of strawberries.")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Total intake for the day: Calories: 2,456.
Fat: 67 g, Carbs: 208 g, Protein: 281 g")
                End If
            ElseIf fitness.FitnessGoal > 2000 Then
                LstDisplayDP.Items.Add("Diet Plan: ")
                LstDisplayDP.Items.Add("")
                If myRnd = 1 Then
                    LstDisplayDP.Items.Add("Breakfast - 2 eggs, 20g of spinach, 24g of
mushrooms, 23g of broccoli, 205g of sautéed sweet potatoes, 1 tablespoon of olive
oil")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack - Apple, 2 tablespoons of peanut
butter")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Lunch - 1 pita, 140g of tuna, chopped red
onion, 1/4 avocado, 1 tablespoon of feta cheese")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack - 56g of cheddar cheese, 92g of
grapes")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Dinner - 140g of baked salmon, 2
tablespoons of olive oil, 82g of rice, 180g of asparagus, 100g of roasted eggplant")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Total intake for the day: Calories: 2,027.
Fat: 22 g, Carbs: 129 g, Protein: 185.8 g")
                ElseIf myRnd = 2 Then
                    LstDisplayDP.Items.Add("Breakfast - 2 slices of whole-grain toast,
2 tablespoons of almond butter, 1 banana, cinnamon")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack - 3/4 cup milk, 20g of spinach, 42g
of protein powder, 123g of frozen blueberries, 1 tablespoon of hemp seeds")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Lunch - 1/2 avocado, 140g of tuna, 75g of
tomatoes 100g of mixed greens")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack - fresh carrot and celery sticks, 2
tablespoons of hummus")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Dinner - 140g of chicken 176g of broccoli,
82g of cooked brown rice, 1 tablespoon of soy sauce")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Total intake for the day:  Calories:
1,913. Fat: 29.3 g, Carbs: 211.9 g, Protein: 153.7 g")
                ElseIf myRnd = 3 Then
                    LstDisplayDP.Items.Add("Breakfast - 200g of yogurt, 74g of
blueberries, 76g of sliced strawberries, 30g of granola")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack - 1 banana, 1 and 1/2 tablespoons of
almond butter")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Lunch - 132g of cooked rice noodles, 141g
of tofu, 2 teaspoons  tamari, 1/2 teaspoon of Sriracha, 2 teaspoons of honey")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack - protein bar of choice")
                    LstDisplayDP.Items.Add("")
```

```
                    LstDisplayDP.Items.Add("Dinner - 3 corn tortillas, 170g of grilled
cod, 1/2 avocado, 2 tablespoons of pico de gallo")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Total intake for the day: Calories: 2005.
Fat: 49 g, Carbs: 201 g, Protein: 124 g")
                ElseIf myRnd = 4 Then
                    LstDisplayDP.Items.Add("Breakfast - 1/2 avocado, 2 slices of
toast, 1 tablespoon of olive oil, 1 egg")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack - 200g of plain Greek yogurt, 125g
of strawberries")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Lunch- 93g of cooked quinoa, 142g of
grilled chicken, 1 tablespoon of olive oil1, 180g of mixed vegetables")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack - 2 squares of dark chocolate, 15-20
almonds")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Dinner - 30g of kidney beans, 103g of
butternut squash, 75g of cooked sweet corn, 1/4 of a jalapeño pepper")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Total intake for the day: Calories: 1945.
Fat: 32 g, Carbs: 185 g, Protein: 108.7 g")
                ElseIf myRnd = 5 Then
                    LstDisplayDP.Items.Add("Breakfast - 80g of steel-cut oats, 1
tablespoonof hemp seeds, 1 tablespoon of flax seeds, 2 tablespoons of dried cherrie")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack - 1/2 bell pepper, 1 cup of carrots,
4 tablespoons of guacamoler")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Lunch - 1 tortilla, 60g of red peppers, 5
slices of grilled zucchini, 84g of fresh mozzarella")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Snack - 170g of chia pudding, 1/2 of a
sliced banana")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Dinner - 2 tablespoons of pesto, 1/2 cup
of whole-wheat or brown-rice penne, 170g of shrimp, 1 tablespoon of Parmesan cheese")
                    LstDisplayDP.Items.Add("")
                    LstDisplayDP.Items.Add("Total intake for the day: Calories: 2,017.
Fat: 22 g, Carbs: 129 g, Protein: 185.8 g")
                End If
            End If
        End While
        connection.Close()
    End Sub
```

Insertion Sort Algorithm (Inside Action Plan Page):

```
Sub InsertionSort()
        Dim fitness(9) As Fitness
        Dim tempFitness As Fitness
        Dim currentPosition As Integer
        Dim newPosition As Integer

        For currentPosition = 1 To 9
            tempFitness = fitness(currentPosition)
            newPosition = currentPosition
            While newPosition > 0 AndAlso fitness(newPosition - 1).FitnessGoal >
tempFitness.FitnessGoal
                fitness(newPosition) = fitness(newPosition - 1)
                newPosition = newPosition - 1
            End While
            fitness(newPosition) = tempFitness
        Next
        MsgBox("Data Sorted")
    End Sub
End Class
```

Help Page:

```
Imports MySql.Data.MySqlClient
Public Class Form7
    Private Sub BtnLoad_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnLoad.Click
        Dim fitness As Fitness
        Dim connection As MySqlConnection
        Dim command As MySqlCommand
        Dim connectionString As String =
"server=localhost;user=root;password=;database=YourFitnessFriend"
        Dim reader As MySqlDataReader
        Dim calories As Integer

        connection = New MySqlConnection(connectionString)
        connection.Open()

        command = New MySqlCommand("SELECT * FROM FitnessData, UserLogin WHERE
FitnessData.userId = UserLogin.userId AND FitnessData.userId = '" & EnterID.Text &
"'")
        command.Connection = connection
        reader = command.ExecuteReader

        While reader.Read()
            fitness.LoseOrGain = reader.Item("LoseOrGain")
            fitness.FitnessGoal = reader.Item("FitnessGoal")
            calories = fitness.FitnessGoal

            If fitness.LoseOrGain = "Lose" Then
                LstDisplayLoseOrGain.Items.Add("To lose weight you must be on a
calorie deficit by eating the same amount of calories as your fitness goal.")
                LstDisplayLoseOrGain.Items.Add("You should aim to eat around " &
calories & " calroies per day.")
                LstDisplayLoseOrGain.Items.Add("")
                LstDisplayLoseOrGain.Items.Add("Top that off with regular exercise,
accountability and patience")
            ElseIf fitness.LoseOrGain = "Gain" Then
                LstDisplayLoseOrGain.Items.Add("To gain weight you must be on a
calorie surplus by eating the same amount of calories as your fitness goal.")
                LstDisplayLoseOrGain.Items.Add("You should aim to eat around " &
calories & " calroies per day.")
                LstDisplayLoseOrGain.Items.Add("")
                LstDisplayLoseOrGain.Items.Add("Top that off with regular exercise,
accountability and patience")
            End If
        End While

        LstDisplayMuscle.Items.Add("To gain muscle you must eat alot of protein")
        LstDisplayMuscle.Items.Add("")
        LstDisplayMuscle.Items.Add("Here are some high protein foods you can")
        LstDisplayMuscle.Items.Add("incorporate into your diet:")
        LstDisplayMuscle.Items.Add("")
        LstDisplayMuscle.Items.Add("- Eggs")
        LstDisplayMuscle.Items.Add("- Chicken")
        LstDisplayMuscle.Items.Add("- Beef")
        LstDisplayMuscle.Items.Add("- Fish")

        LstDisplayGym.Items.Add("Push:")
        LstDisplayGym.Items.Add("")
        LstDisplayGym.Items.Add("Bench Press")
        LstDisplayGym.Items.Add("DB Press")
        LstDisplayGym.Items.Add("Shoulder Press")
```

```
        LstDisplayGym.Items.Add("Tricep Cable Pulldown")
        LstDisplayGym.Items.Add("")
        LstDisplayGym.Items.Add("Pull:")
        LstDisplayGym.Items.Add("Bicep Curls")
        LstDisplayGym.Items.Add("Rows")
        LstDisplayGym.Items.Add("Deadlift")
        LstDisplayGym.Items.Add("Lat Pulldown")
        LstDisplayGym.Items.Add("")
        LstDisplayGym.Items.Add("Legs:")
        LstDisplayGym.Items.Add("")
        LstDisplayGym.Items.Add("Squats")
        LstDisplayGym.Items.Add("Lunge")
        LstDisplayGym.Items.Add("Leg Extention")
        LstDisplayGym.Items.Add("Calf Raises ")

        LstDisplayCardio.Items.Add("Best Low Intensity Cardio Exercises")
        LstDisplayCardio.Items.Add("")
        LstDisplayCardio.Items.Add("Cycling")
        LstDisplayCardio.Items.Add("Walking on Incline")
        LstDisplayCardio.Items.Add("Stair Climbing")

        connection.Close()
    End Sub

    Private Sub BtnHome_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnHome.Click
        LstDisplayLoseOrGain.Items.Clear()
        LstDisplayGym.Items.Clear()
        LstDisplayMuscle.Items.Clear()
        LstDisplayCardio.Items.Clear()
        EnterID.Clear()
        Me.Hide()
        Form3.Show()
    End Sub
End Class
```

Logout Page:

```
Public Class Form4

    Private Sub BtnYes_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnYes.Click
        Me.Hide()
        Form1.Show()
    End Sub

    Private Sub BtnNo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles BtnNo.Click
        Me.Hide()
        Form3.Show()
    End Sub
End Class
```

Database Design and Development SQL Code

Create Database:

```
CREATE DATABASE YourFitnessFriend;
```

Create UserLogin Table:

```
CREATE TABLE UserLogin (
userID int NOT NULL AUTO_INCREMENT,
UsernameData Varchar(20) NOT NULL,
PasswordData Varchar(20) NOT NULL
PRIMARY KEY(userID)
);
```

Create FitnessData Table:

```
CREATE TABLE FitnessData (
id int NOT NULL AUTO_INCREMENT,
userID int NOT NULL,
Gender Varchar(6) NOT NULL CHECK (Gender IN ("Male", "Female")),
Weight Int NOT NULL CHECK (Weight >=45),
Height Int NOT NULL CHECK (Height >=120),
Age Int NOT NULL CHECK (Height >=16),
Exercise Varchar(20) NOT NULL,
LoseOrGain Varchar(4) NOT NULL,
BMR Int NOT NULL,
ExerciseBMR Int NOT NULL,
FitnessGoal Int NOT NULL

PRIMARY KEY(id)
FOREIGN KEY REFERENCES UserLogin(UserID)
);
```

Login SQL:

```
("SELECT * FROM UserLogin WHERE UsernameData='" & EnterUsername.Text & "'
AND PasswordData='" & EnterPassword.Text & "'")
```

Register SQL:

```
("INSERT INTO UserLogin (UsernameData, PasswordData) VALUES('" & Username
& "','" & Password & "')")
```

Verify User SQL:

```
("SELECT UserId FROM UserLogin WHERE UsernameData ='" & UsernameEnter &
"'")
```

Insert Fitness Data SQL:

```
("INSERT INTO FitnessData(UserId, Gender, Weight, Height, Age, Exercise,
LoseOrGain, BMR, ExerciseBMR, FitnessGoal) VALUES('" & Newid & "','" &
Gender & "','" & Weight & "','" & Height & "','" & Age & "','" & Exercise
& "','" & LoseOrGain & "','" & BMR & "','" & ExerciseBMR & "','" &
FitnessGoal & "')")
```

Verify ID SQL:

```
("SELECT * FROM FitnessData, UserLogin WHERE FitnessData.userId =
UserLogin.userId AND FitnessData.userId = '" & EnterID.Text & "'")
```

Research and Development of New Skills and Knowledge

To create my project, I had to research skills that weren't taught as part of the Advanced Higher course. Here are a list of things I had to research:

VB User Interface:

I wanted the users of my program to have a great experience when using my program and wanted to have a successful UI design which contributes to a positive user experience as I feel my users will feel more comfortable while using my application. I wanted it to have a blue background as I felt like it has connotations of health and fitness. I did some research online and found that you can change the VB Forms BackColor in the properties section which allowed me to change the colours. So I implemented this by changing the background colour to be a light blue colour to differentiate my program so it didn't look generic.

VB PasswordChar:

I wanted to make sure my login system was secure so I had to make sure to hide the password field with asterisks for security purposes. I did some research online and found that you can enable UseSystemPasswordChar and set it to True as it will disable people to copy and paste the textbox. I then found out by adding an asterisks (*) in the PasswordChar field, it will replace whatever letter I typed with an asterisks. So when I type a letter it comes with asterisks.

UseSystemPasswordChar	True
PasswordChar	*

VB Add Icon to Button:

I wanted to make my buttons to look attractive so I wanted to add a style to it. I wanted to add icons as it can be a great way to bring essential content to the point. They are a great attention grabber and they will help the users of my program to find and scan content. I had to use royalty free icons online for my project to make sure I did not run into any Copyright infringements. So I went on to a website called Icon Scout to find some copyright free icons for my program.

VB Add Image:

I wanted to add in a logo into my program to ensure that it helps to establish credibility. I wanted it to be on every page except from the logout page. I did some research online and found that you can add images by adding in a PictureBox via the VB tool box. So then I added the logo onto each form except from by logout form.

Sources:

- https://www.techotopia.com/index.php/Visual_Basic_and_Forms#:~:text=To%20change%20the%20background%20color,%2C%20Yellow%2C%20Cyan%20etc

- https://docs.microsoft.com/en-us/office/vba/language/concepts/forms/passwordchar-property

- https://www.homeandlearn.co.uk/NET/nets4p12.html#:~:text=Run%20your%20programme%20and%20Click,and%20then%20click%20File%20%3E%20Open.

- https://iconscout.com/

On Going Testing

What I tested	How did I test it?	What happened?	Notes
Register Account	I entered a username and password greater than 8 characters and It showed an error.	Register account was not successful See Table Below	After Looking at Documentation I managed to get it working
Login Account	I entered the username and password I used to register an account	Login was successful and it directed me to the home page	N/A
Enter Fitness Data	I entered my fitness data but could not get my ID to display. But all of my data appeared in the database.	Entering Fitness Data See Table Below	After Looking at School Recourses and Documentation I managed to get it working and the ID was displayed to the user
Action Plan	I Entered the ID that the program gave to me but threw an error as the fields from the database were not assigning to the record structure	Viewing action plan not successful See Table Below	After Looking at Documentation I managed to get it working and then I could pass values from a database to a record variable
Insertion Sort	I tested this by seeing what is displayed. But I found that it would not sort through the fitness goal and would end up sorting the whole record.	Insertion Sort was successful but didn't sort through the fitness data See Table Below	Later I found that if you declare a temporary variable under the fitness record. It allows you to insert the data into the right order.
Help Page	I entered my ID and it worked as planned as before I fixed up all of the errors regarding declaring variables from database values and connection issues	Test was successful and the help page showed all of the details	N/A
Logout	I clicked the Logout button and pressed the tick	Logout was successful and directed me back to the Login page	N/A

Log of Ongoing Testing Errors

What I tested	Description of Issues	How I Resolved	References
Login System	When I tried to register my login details by using an INSERT statement. It had many errors saying that I could not establish a connection and my data did not enter any table.	I looked up VB documentation on database connection to resolve my issue and get it working.	• https://docs.microsoft.com/en-us/troubleshoot/developer/dotnet/framework/general/open-database-by-sql-server-dotnet-data-provider
SQL Connection + UserID	When I tried to insert my fitness data into my database there was more connection issues that were different to the one previous. I also had to find a way on how to give the UserID to the user after the entered their username.	I had to use multiple SQL statements and connections to allow this to work.	• https://docs.microsoft.com/en-us/troubleshoot/developer/dotnet/framework/general/open-database-by-sql-server-dotnet-data-provider • School Resources and Notes
Reading Data From Database into a Record	When I tried to read the database from the table and select the Fitness Goal and declare it into a record. I ran into many errors regarding passing value from database to a variable.	I had to look up a source on how to do this and implement it.	• https://www.daniweb.com/programming/software-development/threads/388157/passing-value-from-database-to-a-variable

Using an Insertion Sort for an Array of Records	I tried to sort the fitness goals from the fitness data by using an insertion sort algorithm. But I could not find a way on how to sort through an array of records.	I found that if you declare a temporary variable under the fitness record. It allows you to insert the data into the right order.	• N/A
VB Rnd Function	I wanted to give the users of my program a random meal plan depending on their fitness goal. So I wanted to generate a number between 1 and 5. I used the Rnd function but I found that It would display the same number over and over again.	I looked up VB documentation on generating random variables to resolve my issue and get it working so that it generates a different number each time.	• https://docs.microsoft.com/en-us/office/vba/language/reference/user-interface-help/rnd-function

Testing the Solution

Final Test Plan

Persona

My Persona for my program is someone who wants to improve their fitness. They will be an 18 year old male who is in university studying Computing Science and has a part time job at a Tech Start Up so he is very experienced and has no problem navigating around systems. He is around 60kg, 167cm tall and does exercise 3 to 5 times a week. He will be able to register a unique username and password of their choice greater than 8 characters. He will also be able to login with their chosen username and password. He will also be able to easily navigate around my program. He will then be able to enter his fitness data and read his action plan created for him. On top of this, he will be able to view a help page that will help to him reach his desired goal. He has no issues with accessibility and can easily use a keyboard and mouse.

In return the program should be able to store his credentials onto a database, validate all of his inputs to make sure it's all fit for purpose. The program will be able to store all of the user's fitness data onto an external database and finally display an action plan for the user and a help screen. The program should be able to sort through all of the fitness data using the Insertion Sort Algorithm.

For this test the persona will be doing Integrative Testing and will be testing the whole programs performance together. So all of the components of my program will be tested together to see if all systems work with each other.

Test Table

Test Requirements	Description	Test Case	Type of Testing	Expected Output
Register Account	Create New Account	Click on register and register an account Username & password must be greater than 8 characters After registering an account click home	Normal: Greater than 8 characters Exceptional: Letters, symbols and numbers and less than 8 characters Extreme: Equal to 8 characters	Normal: Account will be created and will show message Exceptional: Account will not be created Extreme: Account will be created and will show message
Login Account	Login into New Account	Login using your account details that you used previously	Normal: Exact Details Exceptional: Wrong Details Extreme: Different Capitalisation	Normal: Login will be successful and will direct you to the home page Exceptional: Login will not be successful Extreme: Login will not be successful

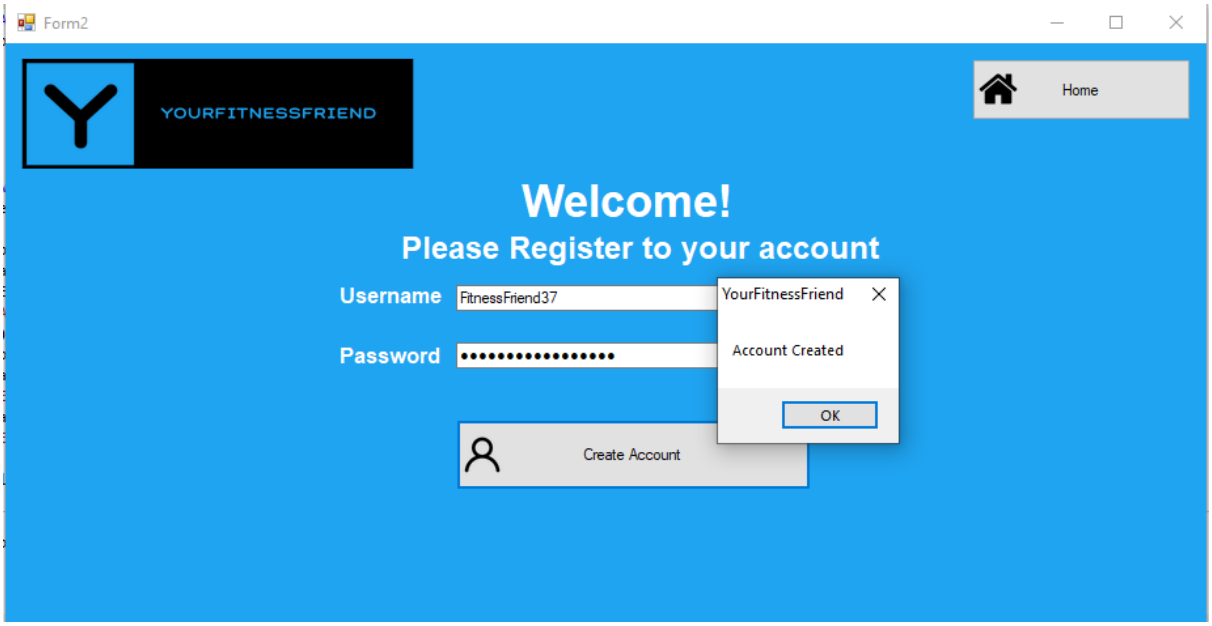
Enter Fitness Data	Enter Username and Calculate Fitness data by Entering Biometric Data. Program will display ID	Enter your username and Enter your data for the program to calculate your fitness goal. Make sure you use your account details that you used previously Age must be greater than or equal to 16 Weight must be greater than or equal to 45 Height must be greater than or equal to 120 After Entering Data Click Home	Normal: Age is greater than 16 Weight is greater than 45 Height is greater than 120 Exceptional: Letters, symbols and numbers and less than 16, 45, 120 Extreme: Equal to 16, 45, 120	Normal: Entering details will be successful and will display a message. The program will display your ID. Exceptional: Entering details will not be successful and will display an error. The Program will not display your ID. Extreme: Entering details will be successful and will display a message. The program will display your ID.

Action Plan	View Action Plan with ID	Enter your ID into the program to view your action plan. Make sure you use your ID that was shown previously After viewing Action Plan click Home	Normal: ID that was assigned to user Exceptional: Letters and Symbols Extreme: Blank	Normal: Showing action plan will be successful and will display a message saying that data has been sorted. Exceptional: Showing action plan will not be successful and will not display message saying that data has been sorted Extreme: Showing action plan will not be successful and will not display message saying that data has been sorted
Help	View Help Page with ID	Enter your ID into the program to view your action plan. Make sure you use your ID that was shown previously. After viewing Help Page click Home	Normal: ID that was assigned to user Exceptional: Letters and Symbols Extreme: Blank	Normal: Showing Help will be successful Exceptional: Showing Help will not be successful Extreme: Showing Help will not be successful
Logout	Click Logout to Exit to the Login Page	On the home page. Click Logout. Click the tick to Logout	Normal: Click the tick to Logout Click the cross to Cancel Exceptional: N/A Extreme: N/A	Logout will be a success and will direct you to the Welcome Page (Login Page) Or Will take you back to the Home Page

Requirements Testing

Register User (Normal):

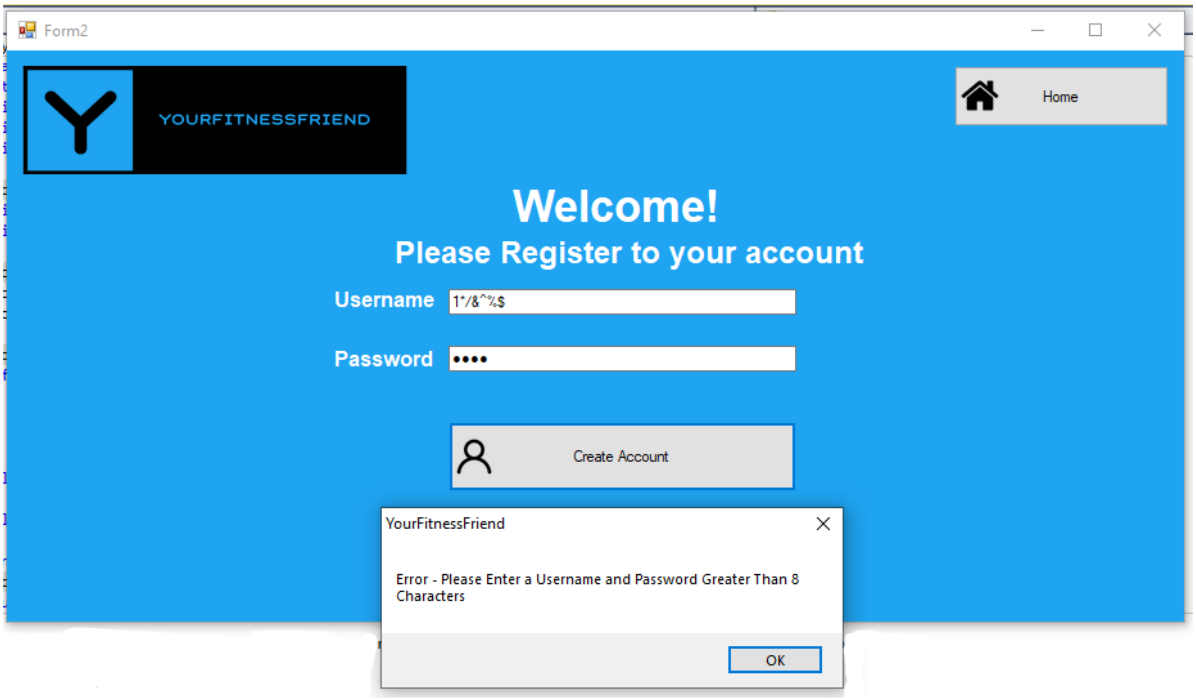
Screenshot of Input:



Database Table Output:

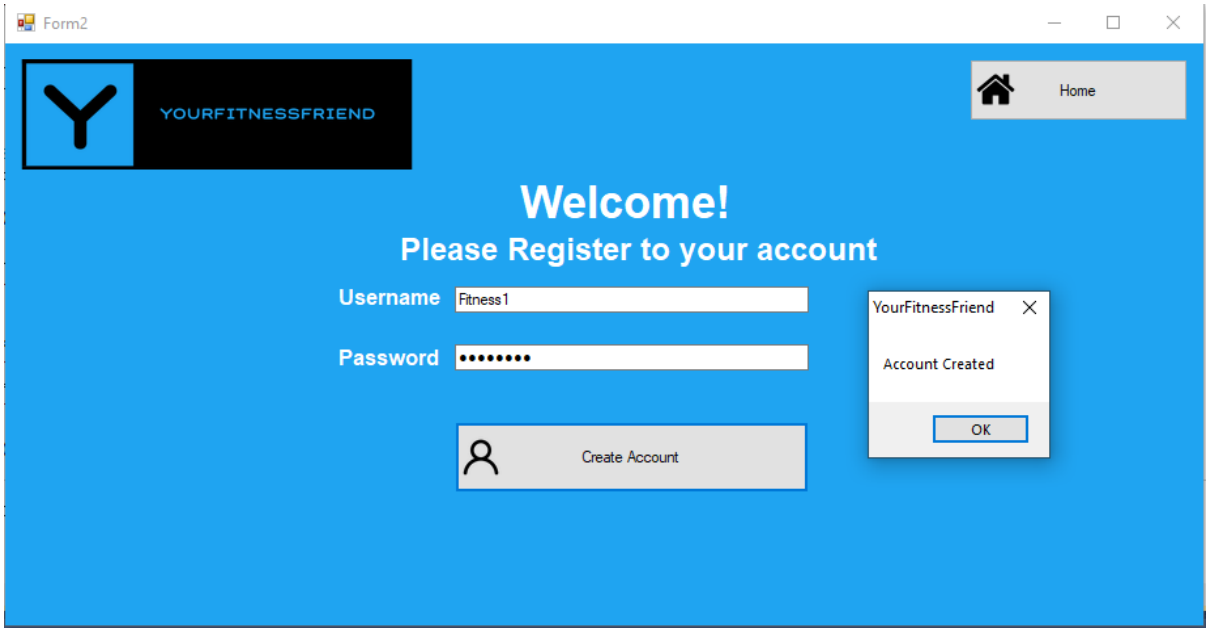
Register User (Exceptional):

Screenshot of Input and Output:



Register User (Extreme):

Screenshot of Input:

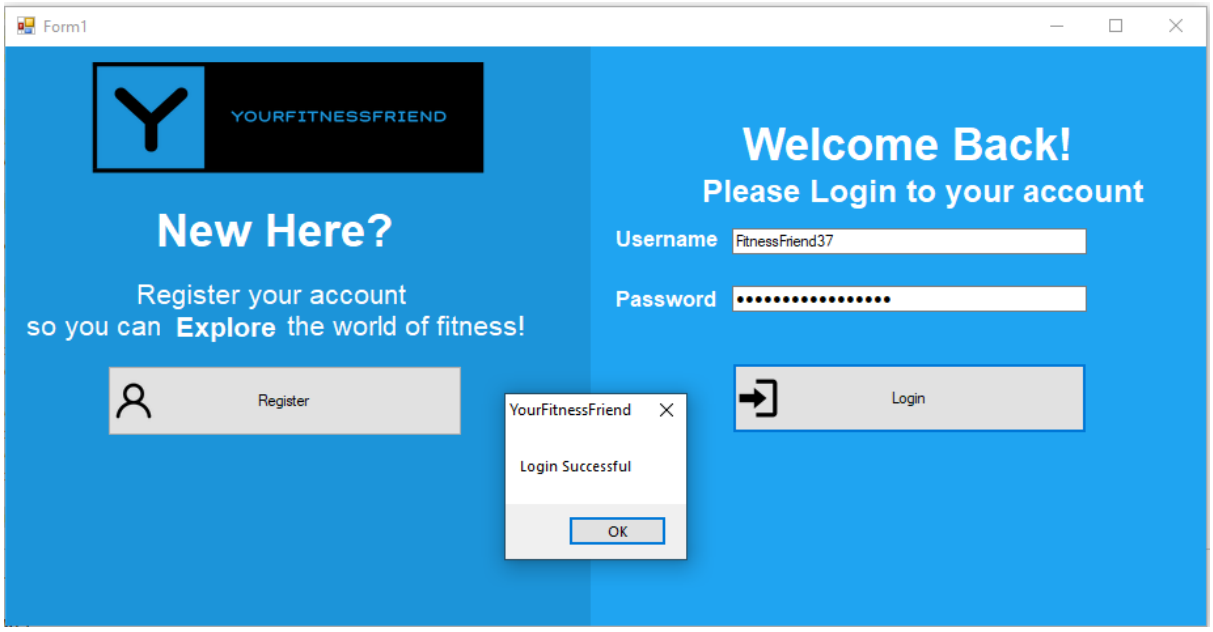


Database Table Output:

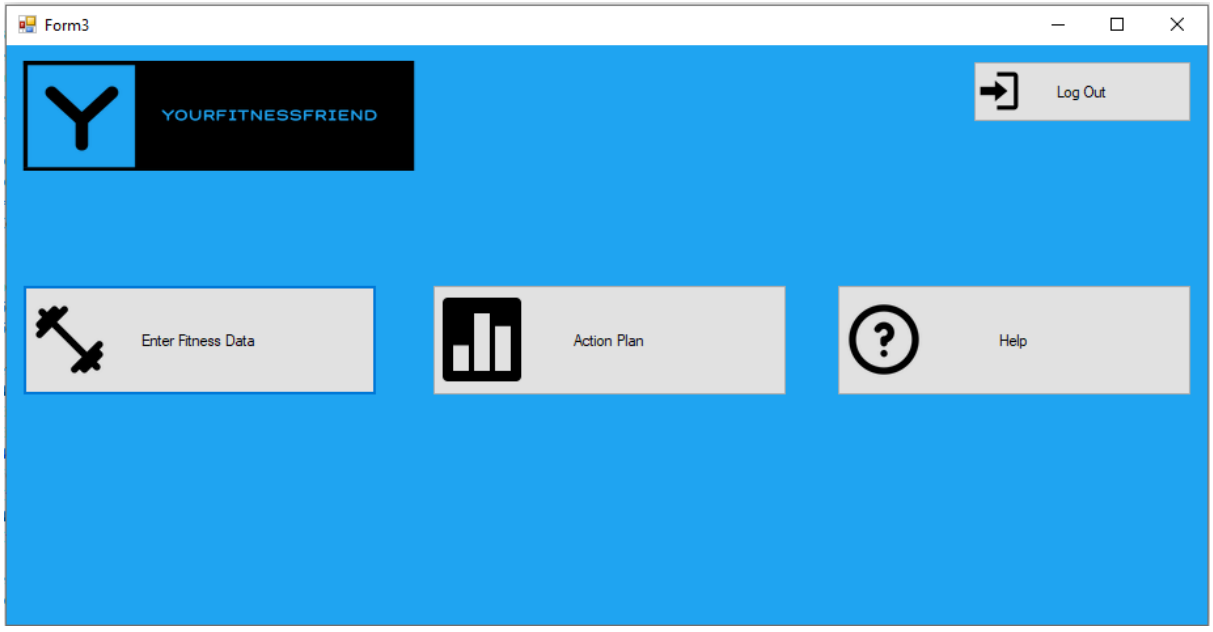
☐	Edit Copy Delete	6	Fitness1	Password

Login (Normal):

Screenshot of Input:

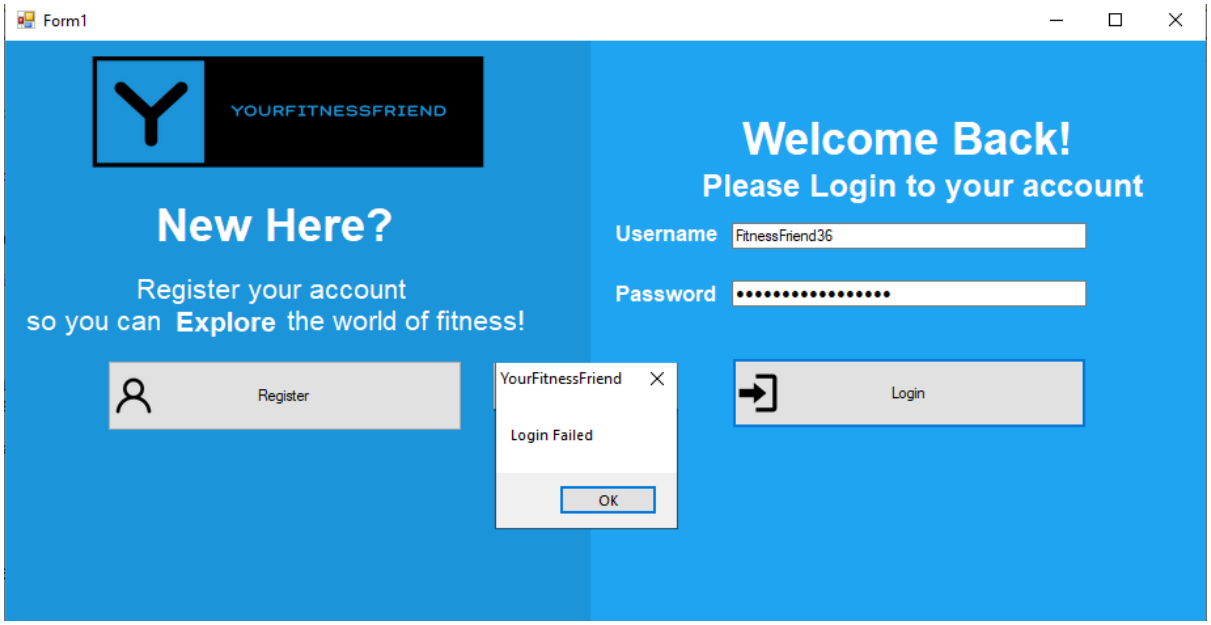


Screenshot of Output:



Login (Exceptional):

Screenshot of Input:

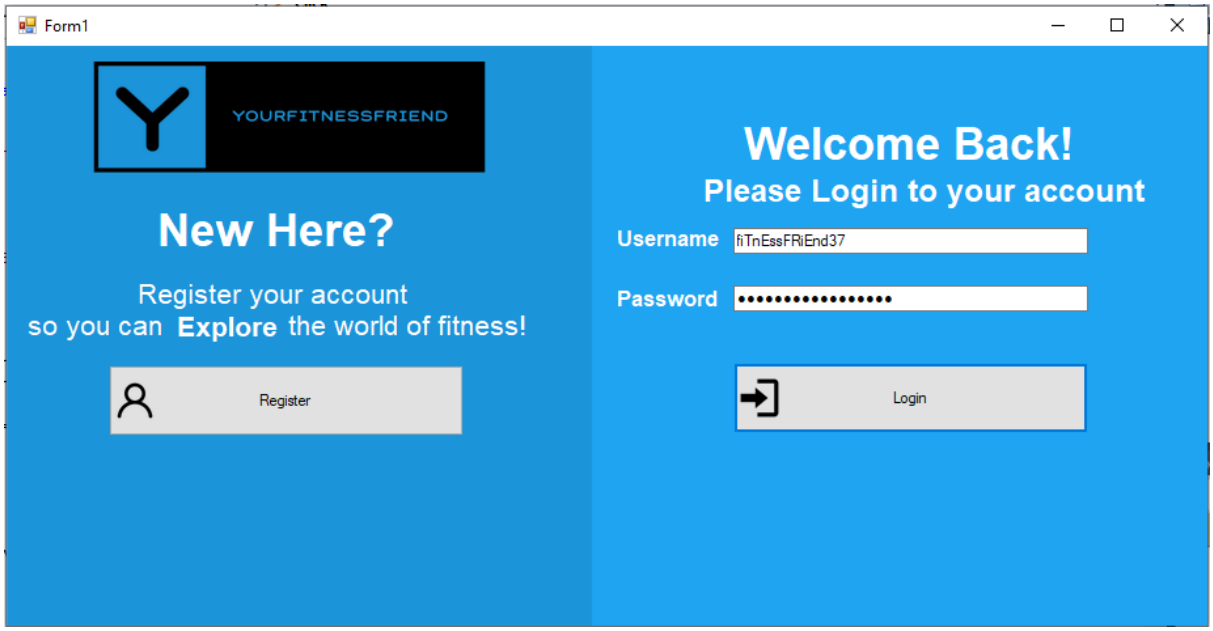


Screenshot of Output:

N/A

Login (Extreme):

Screenshot of Input:

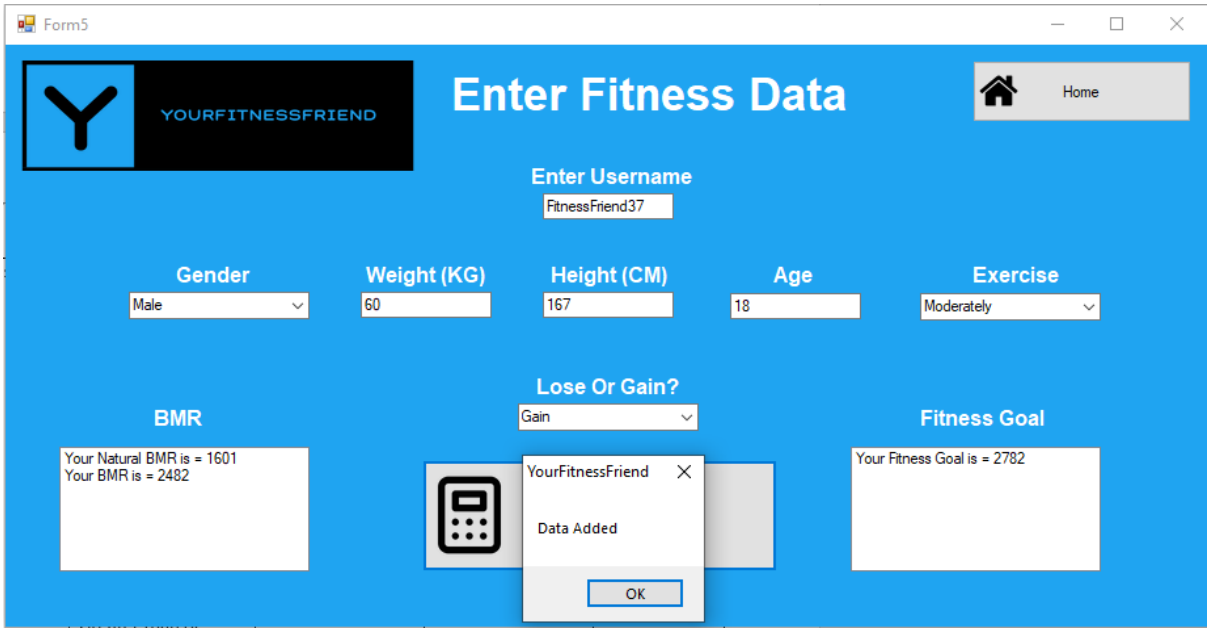


Screenshot of Output:

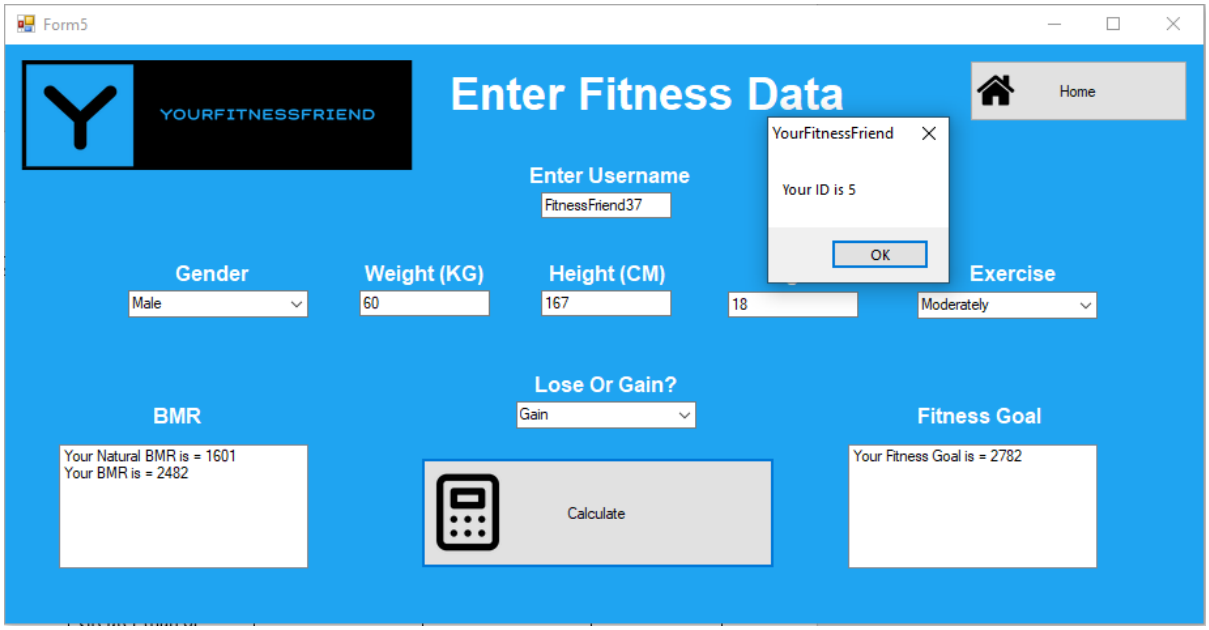
N/A

Enter Fitness Data (Normal):

Screenshot of Input:



Screenshot of Output:

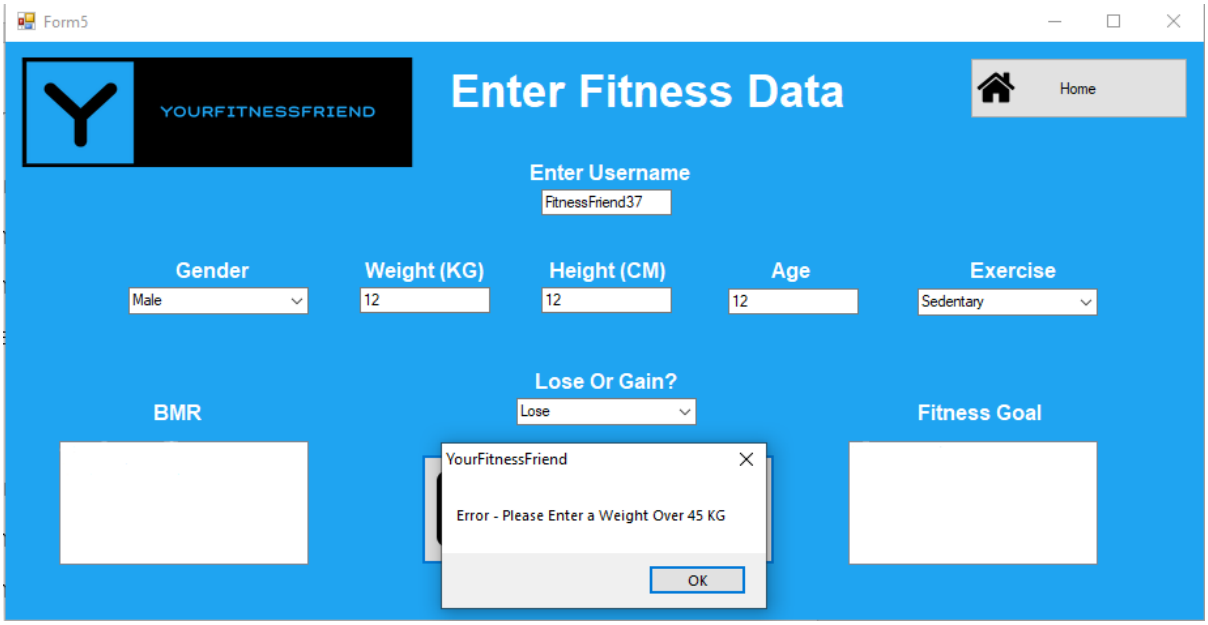


Database Table Output:

☐	Edit Copy Delete	5	5	Male	60	167	18	Moderately	Gain	1601	2482	2782

Enter Fitness Data (Exceptional):

Screenshot of Input:



Screenshot of Output:

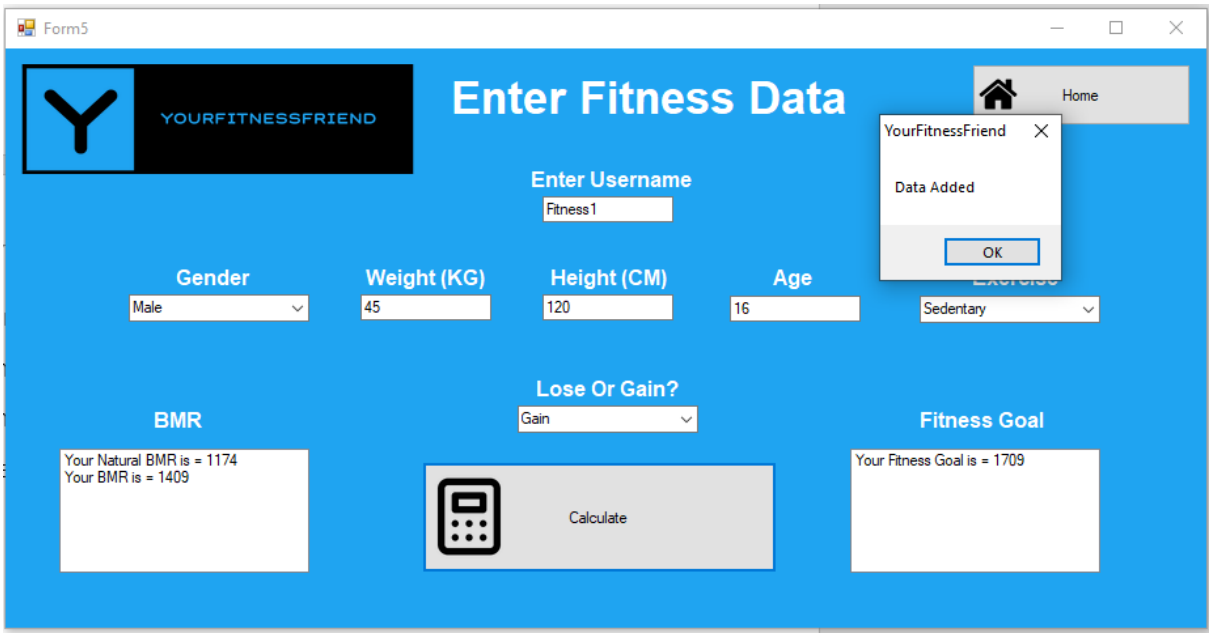
N/A

Database Table Output:

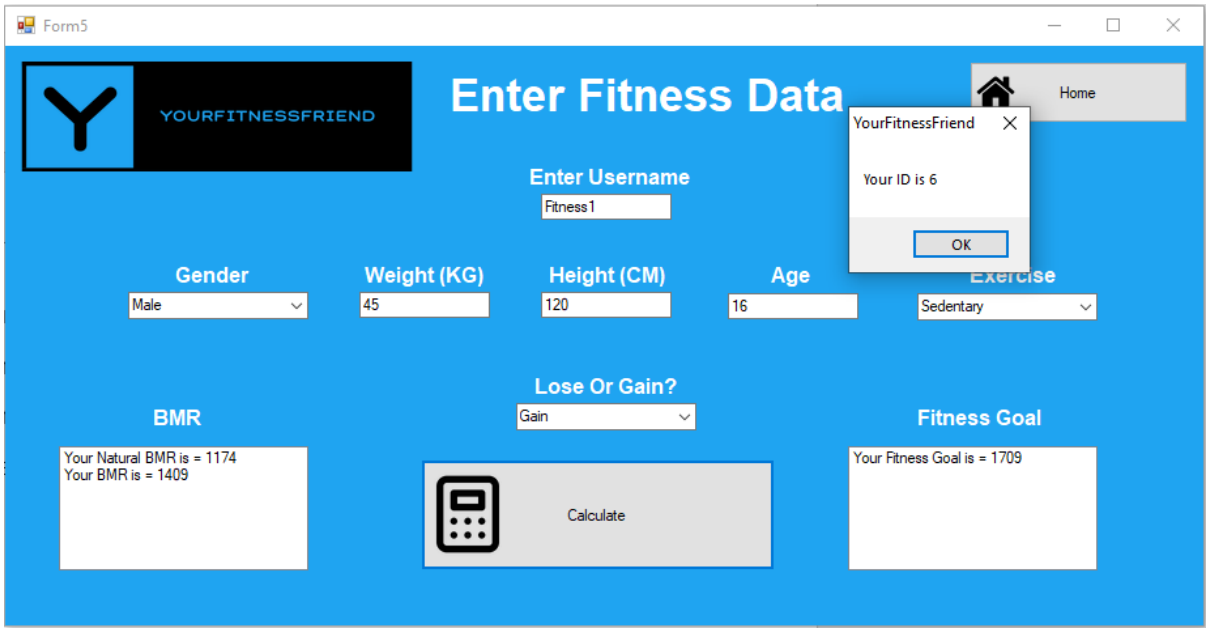
N/A

Enter Fitness Data (Extreme):

Screenshot of Input:



Screenshot of Output:

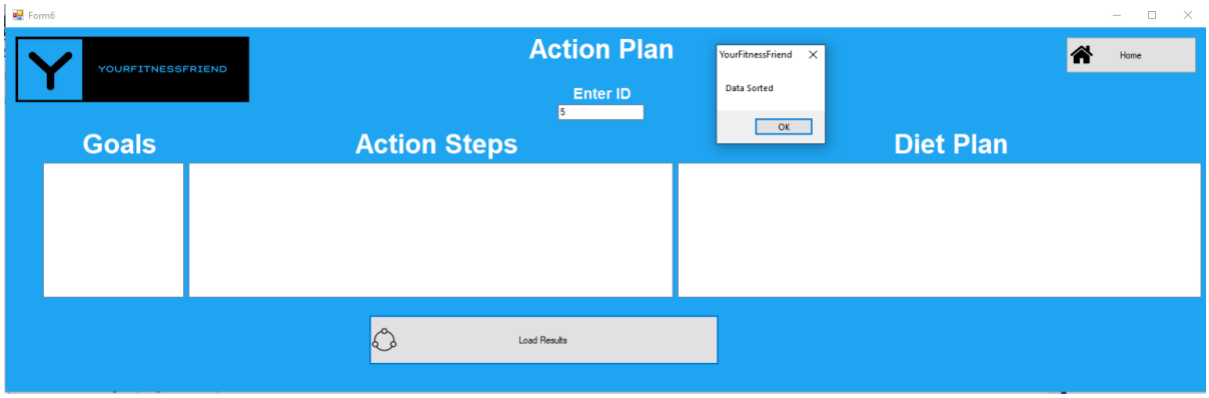


Database Table Output:

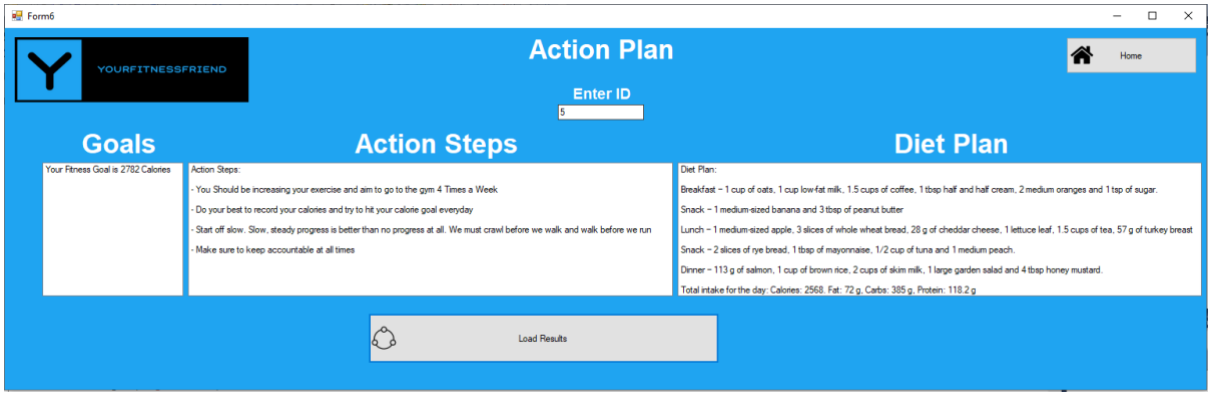
☐	Edit	Copy	Delete	6	6	Male	45	120	16	Sedentary	Gain	1174	1409	1709

Action Plan (Normal):

Screenshot of Input:



Screenshot of Output:



Action Plan (Exceptional):

Screenshot of Input:

Screenshot of Output:

N/A

Action Plan (Extreme):

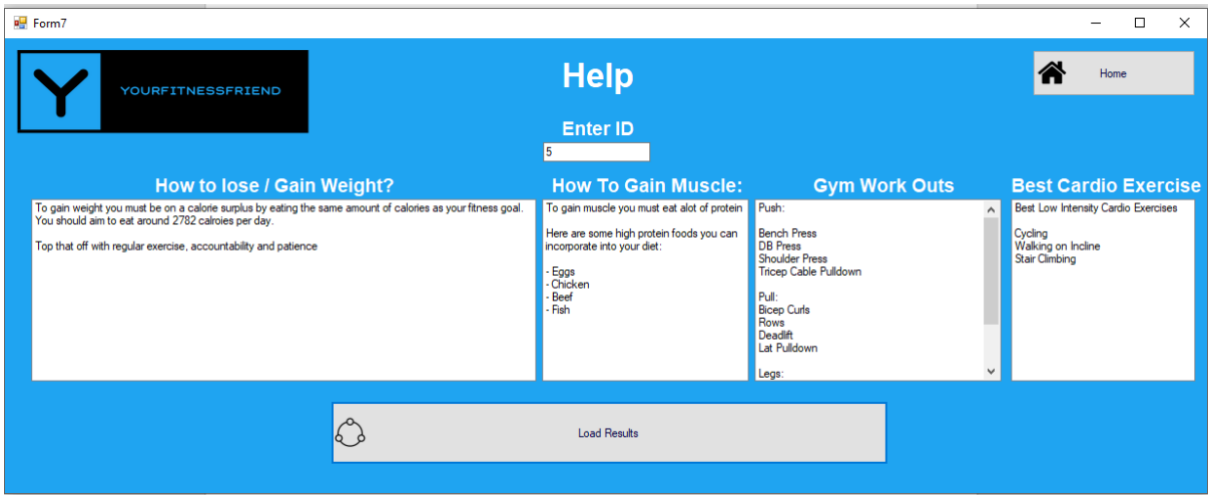
Screenshot of Input:

Screenshot of Output:

N/A

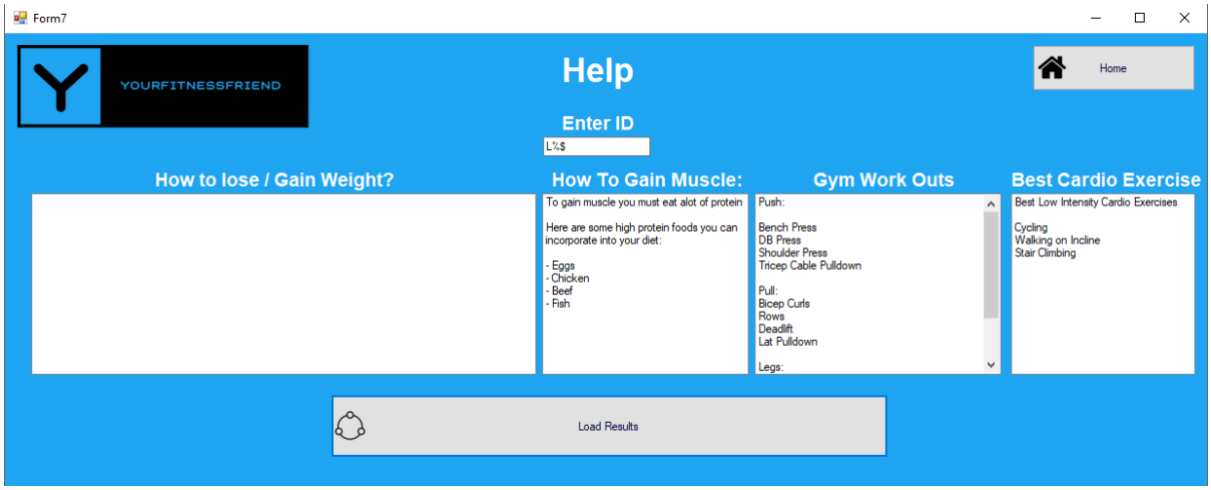
Help (Normal):

Screenshot of Input:



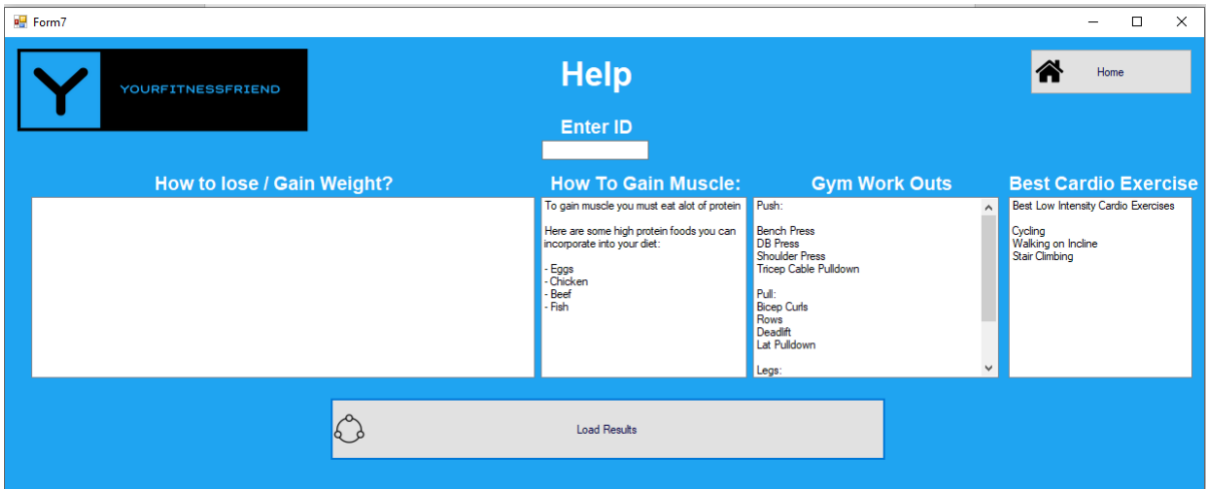
Help (Exceptional):

Screenshot of Input:



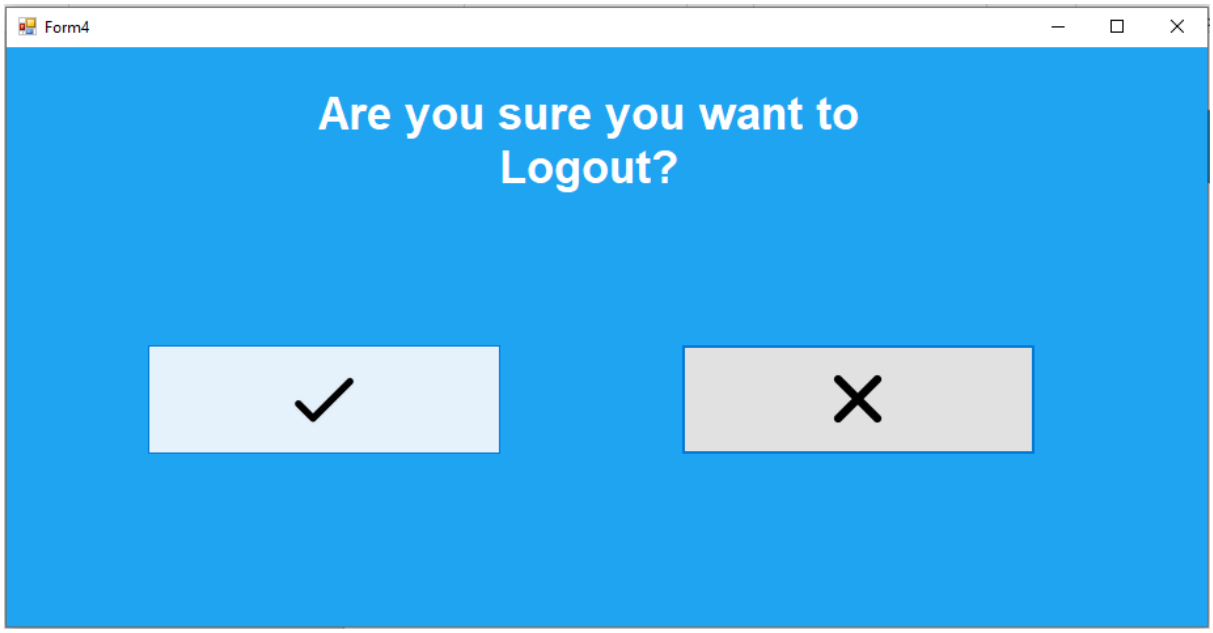
Help (Extreme):

Screenshot of Input:

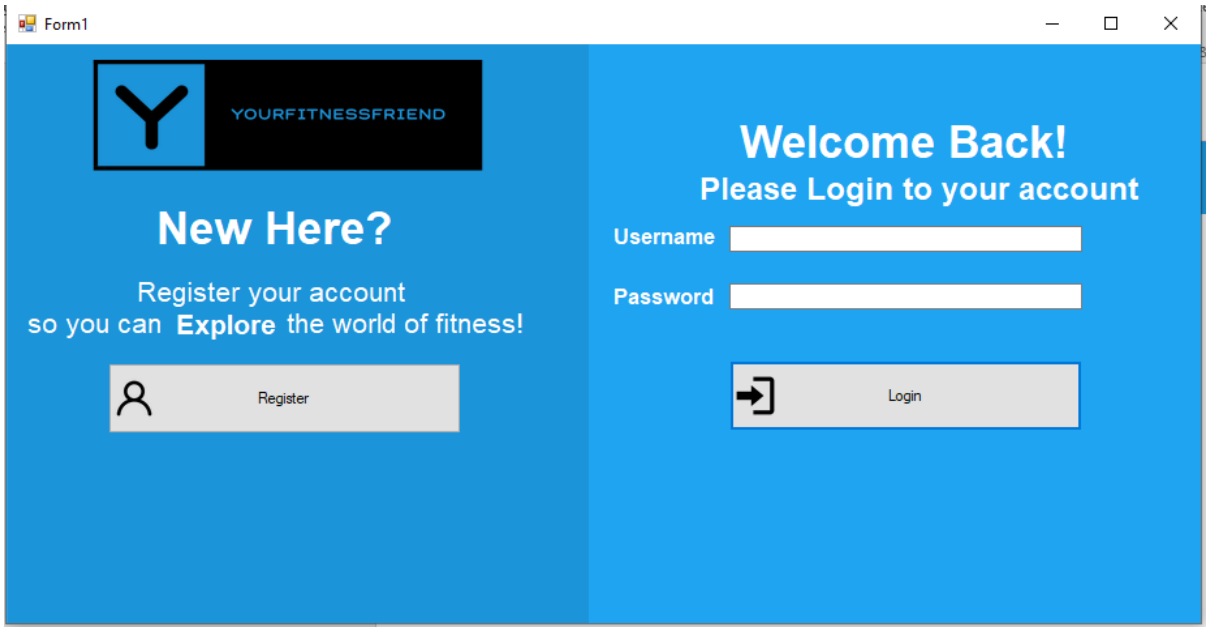


Logout (Normal) [Tick]:

Screenshot of Input:

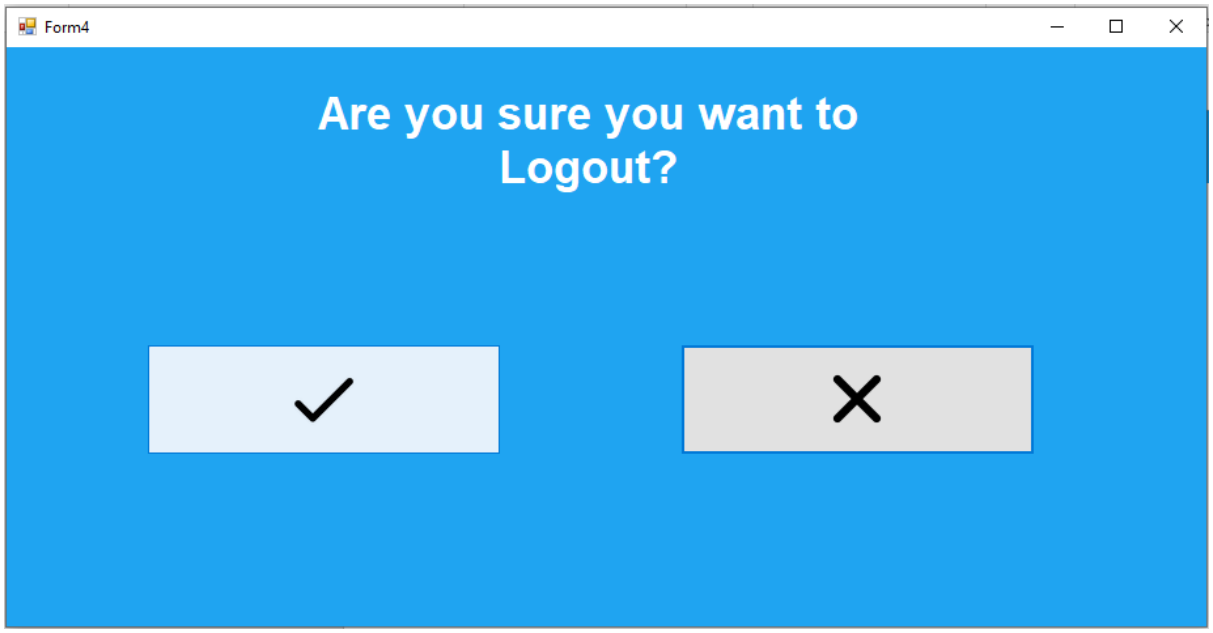


Screenshot of Output:

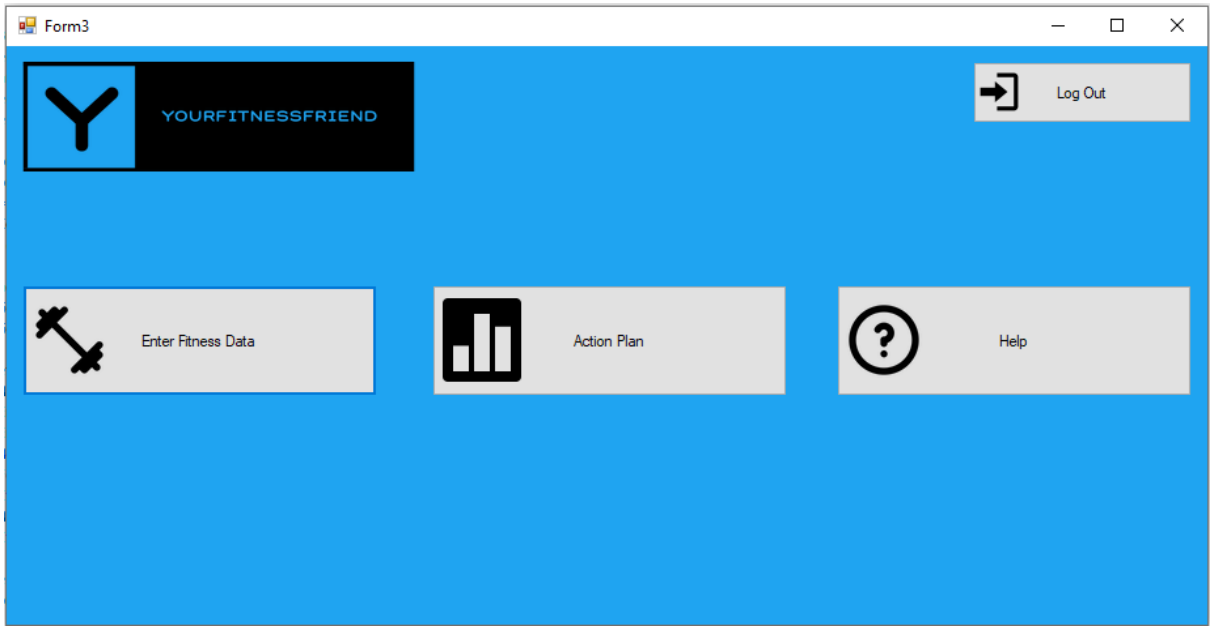


Logout (Normal) [Cross]:

Screenshot of Input:



Screenshot of Output:



Test Results

Test Case	Actual Result
Register	
Register (Normal)	Account is created with unique username and password
Register (Exceptional)	Account is not created and shows error
Register (Extreme)	Account is created with unique username and password
Login	
Login (Normal)	Login is successful and directs to home page
Login (Exceptional)	Login failed and shows error
Login (Extreme)	Login failed. No error shown
Enter Fitness Data	
Enter Fitness Data (Normal)	Data is entered successfully and ID is shown and data is sent to Database
Enter Fitness Data (Exceptional)	Data is not entered and shows error. Data is not sent to Database
Enter Fitness Data (Extreme)	Data is entered successfully and ID is shown and data is sent to Database
Action Plan	
Action Plan (Normal)	Action Plan is shown
Action Plan (Exceptional)	Action Plan is not shown
Action Plan (Extreme)	Action Plan is not shown
Help	
Help (Normal)	Help Page is shown
Help (Exceptional)	Help Page is partly shown. How to Lose / Gain Weight is not shown
Help (Extreme)	Help Page is partly shown. How to Lose / Gain Weight is not shown
Logout	
Logout (Normal) [TICK]	Logout successful and user is directed to Login Page
Logout (Normal) [CROSS]	Logout is not successful and user is directed to Home page

Evaluation of the solution

Fitness for Purpose

After testing my project I now know and confirm that my project meets all of the end user and functional requirements.

The results of my testing show that users are able to register a unique username and password of their choice greater than 8 characters, login with their chosen username and password, navigate around my program, enter their fitness data, read their action plan created for them and will also be able to view a help page that will help to reach their desired goal. After testing, it showed that my program can store the user's login details and fitness data in an external database.

All inputs have been validated to ensure they are fit for purpose and match my requirements. The program will also display the action plan and help page.

My program successfully makes use of the AH concepts we learned in class: The Insertion Sort Algorithm and an Array of Objects. It reads the fitness data from the database and sorts them in order and displays the correct Fitness Goal. The Class was called in an external script and was implemented throughout the program. Throughout testing it showed that both the sort algorithm and the class worked and met the requirements.

My program does the integration of the SQL database effectively for my implementation. The connection string was created and the data was inserted and read effectively each time and stored all of my data while giving each row a unique identifier by using auto increment.

After testing with the test cases it showed that my persona was able to successfully complete all tasks given to them. I also managed to complete all of the tests in my test plan and then show evidence for each of the tests. I am confident that my project meets all of the requirements listed in the end user and functional requirements.

Requirements Check List

End User Requirements:	**Completed**
The users for my system will be able to register a unique username and password of their choice.	✓
The user will be allowed to login using their own chosen username and password.	✓
The user will be able to easily navigate throughout my program.	✓
The user will be able to input their own fitness data into their program and be able to see their action plan	✓
The user will also be able to access a help screen for tips to reach their desired goal	✓
Functional Requirements:	
My program will be able to store user credentials in an external database.	✓
My program will be able to validate all user inputs to make sure they are fit for purpose.	✓
My program will store the users data in an external database	✓
My program will display the user's action plan and help page.	✓

Maintainability

I believe that my solution is maintainable. I have used meaningful variable names for all of my VB class and record structures and also for the SQL database fields. This allows for Improved Code integration, clarity and consistency. Making meaningful distinctions in my code allows me to identify what each variable does. Here is a screenshot showing some of the variables names I used in my VB Class.

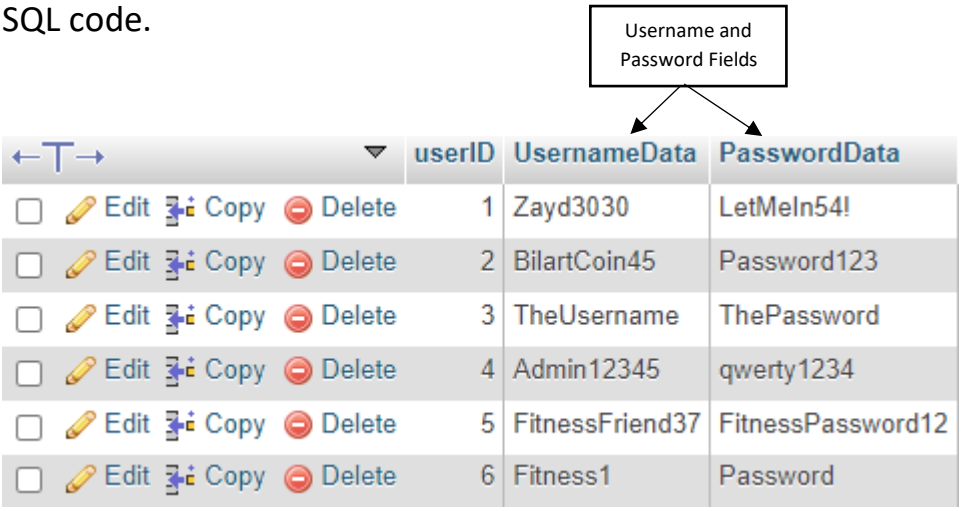
```
Public Class FitnessData
    Public id As Integer
    Public Gender As String
    Public Weight As Integer
    Public Height As Integer
    Public Age As Integer
    Public Exercise As String
    Public LoseOrGain As String
    Public BMR As Integer
    Public ExerciseBMR As Integer
    Public FitnessGoal As Integer
```

For example:

The variable name BMR is distinguishable from ExerciseBMR

BMR relates to the raw BMR which has been calculated for the user. The ExerciseBMR relates to the updated BMR after how much exercise the user has done is been accounted for. Having distinguishable names allows me to recognise and identify different sections of my code. Since it's held in a class this also means that I can reuse my code for other future developments or updates required, it will be easier as it has held strong maintenance.

Here is a screenshot showing some of the database field's names I used in my SQL code.



I made sure to use the names UsernameData and PasswordData to ensure I did not run into any errors in my program code as having a field name "Password" will have its own authentication and will flag up an error.

I also used a lot of white space in my code into split the code into several sections and I have also called in sub procedures so that my code is clean and can perform the right procedures.

Here is a section of my code which shows all of that.

```
Public Class Form6
    Private Sub BtnHome_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnHome.Click
        LstDisplayGoal.Items.Clear()
        LstDisplayAS.Items.Clear()
        LstDisplayDP.Items.Clear()
        EnterID.Clear()
        Me.Hide()
        Form3.Show()
    End Sub

    Private Sub BtnLoad_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnLoad.Click
        Dim fitness As Fitness
        Dim connection As MySqlConnection
        Dim command As MySqlCommand
        Dim connectionString As String = "server=localhost;user=root;password=;database=YourFitnessFriend"
        Dim reader As MySqlDataReader
        Dim myRnd As Integer
        Dim days As Integer

        connection = New MySqlConnection(connectionString)
        connection.Open()

        command = New MySqlCommand("SELECT * FROM FitnessData, UserLogin WHERE FitnessData.userId = UserLogin.userId AND FitnessData.userId = '" & EnterID.Text & "'")
        command.Connection = connection
        reader = command.ExecuteReader

        While reader.Read()
            fitness.FitnessGoal = reader.Item("FitnessGoal")
            Call InsertionSort()
            LstDisplayGoal.Items.Add("Your Fitness Goal is " & fitness.FitnessGoal & " Calories ")

            fitness.Exercise = reader.Item("Exercise")

            If fitness.Exercise = "Sedentary" Then
                days = 3
            ElseIf fitness.Exercise = "Lightly" Then
                days = 3
            ElseIf fitness.Exercise = "Moderately" Then
                days = 4
            ElseIf fitness.Exercise = "VeryActive" Then
                days = 5
            ElseIf fitness.Exercise = "ExtraActive" Then
                days = 6
            End If
```

Multiple Sub Routines

Calls in Insertion Sort Sub routine

White Space to separate code into sections for readability

Robustness

All of my inputs for my program are robust as every input is validated to ensure the user encounters no errors when using my program. When registering an account I implemented this code to ensure that the username and password is greater than 8 characters, is not under 8 characters and is not left blank.

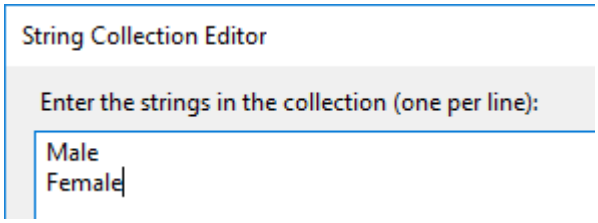
```
connection.Open()
If Len(Username) >= 8 Or Len(Password) >= 8 Then
    command.ExecuteNonQuery()
    MsgBox("Account Created")
    Me.Hide()
    Form1.Show()
ElseIf Username <> "" Or Password <> "" Then
    MsgBox("Error - Please Enter a Username and Password Greater Than 8 Characters")
ElseIf Username = "" Or Password = "" Then
    MsgBox("Error - Please Enter a Username and Password")
End If
connection.Close()
```

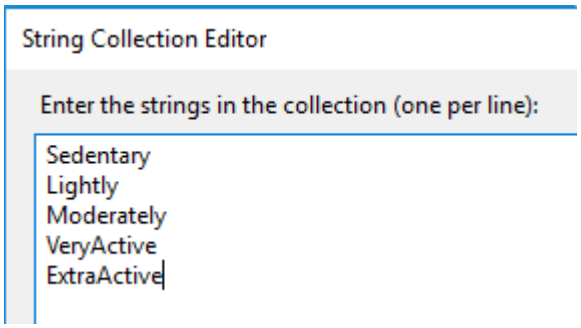
Also when logging into an account I implemented this code to ensure that the username and password is converted to a string to make sure I do not encounter any errors and made sure it matched the username and password entered into the text box matched the one in the database.

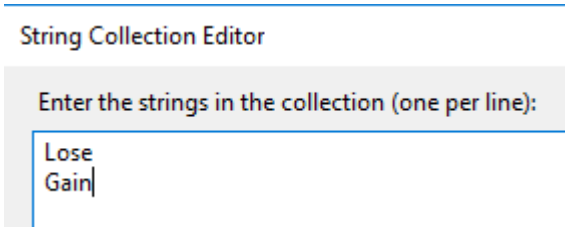
```
While reader.Read()
    Dim GetUsername As String = reader.GetString(1).ToString
    Dim GetPassword As String = reader.GetString(2).ToString

    If GetUsername = EnterUsername.Text And GetPassword = EnterPassword.Text Then
        MsgBox("Login Successful")
        EnterUsername.Clear()
        EnterPassword.Clear()
        Me.Hide()
        Form3.Show()
    ElseIf GetUsername <> EnterUsername.Text And GetPassword <> EnterPassword.Text Then
        MsgBox("Login Failed")
    End If

End While
```

When entering data I wanted to ensure that my program maintained robustness so I added input validation to every input. I added combo boxes to reduce the user's choice to make sure they are no errors and showed error messages to tell the user if they don't meet the requirement.

```
If Person1.Weight < 45 Then
    MsgBox("Error - Please Enter a Weight Over 45 KG")
ElseIf Person1.Height < 120 Then
    MsgBox("Error - Please Enter a Height Over 120 CM")
ElseIf Person1.Age < 16 Then
    MsgBox("Error - Please Enter an Age Over 16")
Else
    command.ExecuteNonQuery()
    MsgBox("Data Added")
    MsgBox("Your ID is " & Newid)
```







Final Evaluation

I am very confident that my implemented project matches my design and requirements as shown from my Testing and Evaluation. I have made sure that my full completed project was completed by the deadline of the 4th of March 2022. I believe, I have not run into any copyright issues by using royalty free icons and creating my logo on a free logo making website. Also, when it came to the name of my project “YourFitnessFriend”, I had to comply with the Copyright, Design and Patents Act 1988 so it wasn’t similar to “MyFitnessPal” by Under Armour. I made sure that there was no costs involved during the course of my project as all of the software and materials are supplied my school. I made sure to do my programming with the Visual Basic Language within the Visual Studio 2010 software as I have several years of experience using this and used EasyPHP as my Database Server while coding in MySQL. As I believe that this was most suitable for my project. Therefore I am confident that my project matches the design and requirements as shown by my evaluation of my project.