

National College of Ireland

Business Information System

BIS

2021/2022

Brandon Kinsella

X18510469

X18510469@student.ncirl.ie

Software Project

Technical Report

Contents

Executive Summary	6
1.0 Introduction	6
1.1. Background	6
1.2. Aims.....	6
1.3. Technology.....	7
1.4. Structure	7
2.0 System.....	8
2.1. Requirements.....	8
2.1.1. Functional Requirements.....	8
User	8
Registration Page	8
Login.....	8
Training Plan	9
Diet Plan	9
Nutrition.....	9
2.1.1.1. Use Case Diagram	Error! Bookmark not defined.
2.1.1.2. Requirement 1: User Registration/Login	Error! Bookmark not defined.
2.1.1.3. Description & Priority.....	Error! Bookmark not defined.
2.1.1.4. Use Case	10
Use Case Diagram	10
2.1.1.5. Requirement 1: User Registration/Login	10
2.1.1.6. Description & Priority.....	11
2.1.1.7. Use Case	Error! Bookmark not defined.
Requirement 1: User Registration/Login	11
Scope.....	11

Description	11
Use Case Diagram	11
Flow Description	11
Precondition.....	11
Activation	12
Main flow	12
Alternate flow	12
Exceptional flow	12
Termination.....	12
Post condition	12
2.1.2. Data Requirements	12
User Data	12
Programme Data	13
2.1.3. User Requirements	13
Unregistered User	13
Registered User	13
2.1.3.1. Requirement 1: User Registration/Login	Error! Bookmark not defined.
2.1.3.2. Description & Priority.....	Error! Bookmark not defined.
2.1.3.3. Use Case	Error! Bookmark not defined.
1. Requirement 2: Dashboard.....	13
Scope.....	13
Description	13
Use Case Diagram	13
Flow Description	14
Precondition.....	14
Activation	14
Main flow	14
Alternate flow	14
Exceptional flow	14
Termination.....	15
Post condition	15
2.1.4. Data Requirements	15
User Data	15
Programme Data	15
2. Requirement 3: Training Plan	15
Scope.....	15

Description	15
Use Case Diagram	16
Flow Description	16
Precondition.....	16
Activation	16
Main flow	16
Alternate flow	16
Exceptional flow	16
Termination.....	17
Post condition	17
2.1.5. Data Requirements	17
User Data	17
Programme Data	17
3. Requirement 4: Diet Plan	17
Scope.....	17
Description	17
Use Case Diagram	18
Flow Description	18
Precondition.....	18
Activation	18
Main flow	18
Alternate flow	18
Exceptional flow.....	18
Termination.....	19
Post condition	19
2.1.6. Data Requirements	19
User Data	19
Programme Data	19
4. Requirement 5: Calorie Counter	19
Scope.....	19
Description	19
Use Case Diagram	20
Flow Description	20
Precondition.....	20
Activation	20
Main flow	20

Alternate flow	20
Exceptional flow.....	20
Termination.....	21
Post condition	21
2.1.7. Data Requirements	21
User Data	21
Programme Data	21
5. Requirement 6: Results.....	21
Scope.....	21
Description	21
Use Case Diagram	22
Flow Description	22
Precondition.....	22
Activation	22
Main flow	22
Alternate flow	22
Exceptional flow.....	22
Termination.....	22
Post condition	23
6. Requirement 7: About us/Nutrition.....	23
Scope.....	23
Description	23
Use Case Diagram	23
Flow Description	23
Precondition.....	23
Activation	23
Main flow	23
Alternate flow	24
Exceptional flow.....	24
Termination.....	24
Post condition	24
2.1.8. Environmental Requirements	24
2.1.9. Usability Requirements	24
2.2. Design & Architecture	25
2.3. Implementation	26
Register New User.....	26

User Login 1	27
User Login 2	27
2.4. Graphical User Interface (GUI)	28
Registration Page	28
Login Screen	29
2.5. Testing	Error! Bookmark not defined.
2.6. Evaluation	Error! Bookmark not defined.
3.0 Conclusions	36
4.0 Further Development or Research	36
5.0 References	37
6.0 Appendices.....	37
6.1. Project Proposal	37
1.0 Objectives.....	38
2.0 Background	38
3.0 State of the Art.....	39
4.0 Technical Approach.....	39
5.0 Technical Details	39
6.0 Special Resources Required	40
7.0 Project Plan	40
Steps:.....	40
6.2. Reflective Journals	41
September 2021.....	41
October 2021	41
November 2021	41
December 2021.....	41

Executive Summary

As a part of my software project, I made a lot of research into many areas of app development and many topics to pick from, I came to my final decision that a gym application would be a great suit for me as its both enjoyable and suits my lifestyle The gym application purpose will be to suit all ages from 16+ to suit and tailor there training with their diet plan, the app will cater to all levels of training and involve a polar API and results for which will be done on a weekly. The application is tailored to help all levels of training, providing health and fitness tips for a healthy life. From my studies it is essential for all individuals to have a active healthy lifestyle to reduce levels of obesity and other medical problems.

Through user-generated content, the app provides an endless supply of workout plans and health and fitness tips, as well as a community where users may advise and assist one another with their health and fitness objectives. I will describe the technology for which I am using, and the use cases involved in my project also giving a brief description of the design and the tools involved in my project.

1.0 Introduction

1.1. Background

I choose to do a gym application as it is something that would help me and many more people in daily life and, I have an interest in the process behind the building of applications/games. I got the idea from being asked common questions from friends and seeing it posted on social media, the idea was then finalized when I researched other gym applications and didn't not see another application like mine.

As there is not much competition in regards my gym application, I seen it was a chance to develop an application that catered for all levels for which I had not seen when research and also testing other applications.

1.2. Aims

This project aims to help all levels of training and all ages as it makes it easier to live a healthy lifestyle and be able to set your workout and see your goals while training. I am hoping that I can add to the application after this project to release in years to come as I can see a niche market.

The main objective of the project is also for myself to progress with the different obstacles for which I have face while building the application and how I will be able to fix these problems when building another application.

As it was my first ever fully build on an application for myself, I aim to have majority of the application functionally working to the best of its ability so that it can help others in the future.

1.3. Technology

I used visual studios to design and create the forms and using visual studios forms to develop my application. Microsoft Visual Studio is an integrated development environment. It's used to make websites, web apps, online services, and mobile apps, among other things. Microsoft software development platforms such as Windows API, Windows Forms, and others are used by Visual Studio. I am hoping that I can use android emulator to deploy my application and during the implementation of the Web APIs for certain features within the application for which cause a lot of problems in my application when trying to implement both polar heart rate API and google maps API. As for my database connected, I used MS access as for my device was having trouble when trying to use MySQL as the port had been changed and caused myself many difficulties with both the login and registration of the application. Microsoft Access is a relational database management application that allows you to save information for reference, reporting, and analysis but on my laptop the version to allow Ms access directly work with Visual Studios did not match the bit of the Ms access version. After weeks of trying to use other applications on my laptop I was unable to come up with another solution other than using a connection string within my code to connect the data base to visual studio myself.

1.4. Structure

The structure of the document will consist of the System and addressing the Functional and non-functional requirements, use cases and the flows involved in the project. All the requirements including data, usability, testing will also be in the document and will then conclude on the progress of my project and where I am in my progress.

2.0 System

2.1. Requirements

The user should be able to navigate the application after 10 mins of navigating each off the commands as the application is straight forwards and very feasible to all ages. If the user has a problem, there will be a query function to report a problem.

2.1.1. Functional Requirements

User

1. All users of smart phones or tablets should have the ability to create an account and have full use of the application.
2. The application shall store user details and plans but does not have the ability to share user information.
3. Login and register are mandatory for use of the Application.

Registration Page

1. On the opening of the registration GUI all users should be easily able to set up an account.
2. The system shall ask for username.
3. The system shall ask for password.
4. The system shall ask for password confirmation.
5. The system should not accept no entries for username or password.
6. The system shall notify user of false entries.
7. The system shall confirm when user is registered.

Login

1. On the opening of the login GUI all users should be easily able to login if already a registered applicant.
2. The system shall prompt if nothing is entered on login
3. The system shall not accept users that are not registered.

Training Plan

1. On Opening of dashboard users shall be able to interact with the Training Plan button to access the GUI Training Plan.
2. On loading of the GUI all information and edited workouts and new workouts should be loaded by the data base in relation to the account of the user.
3. All users should be able to see their GUI results from the weekly chart.

Diet Plan

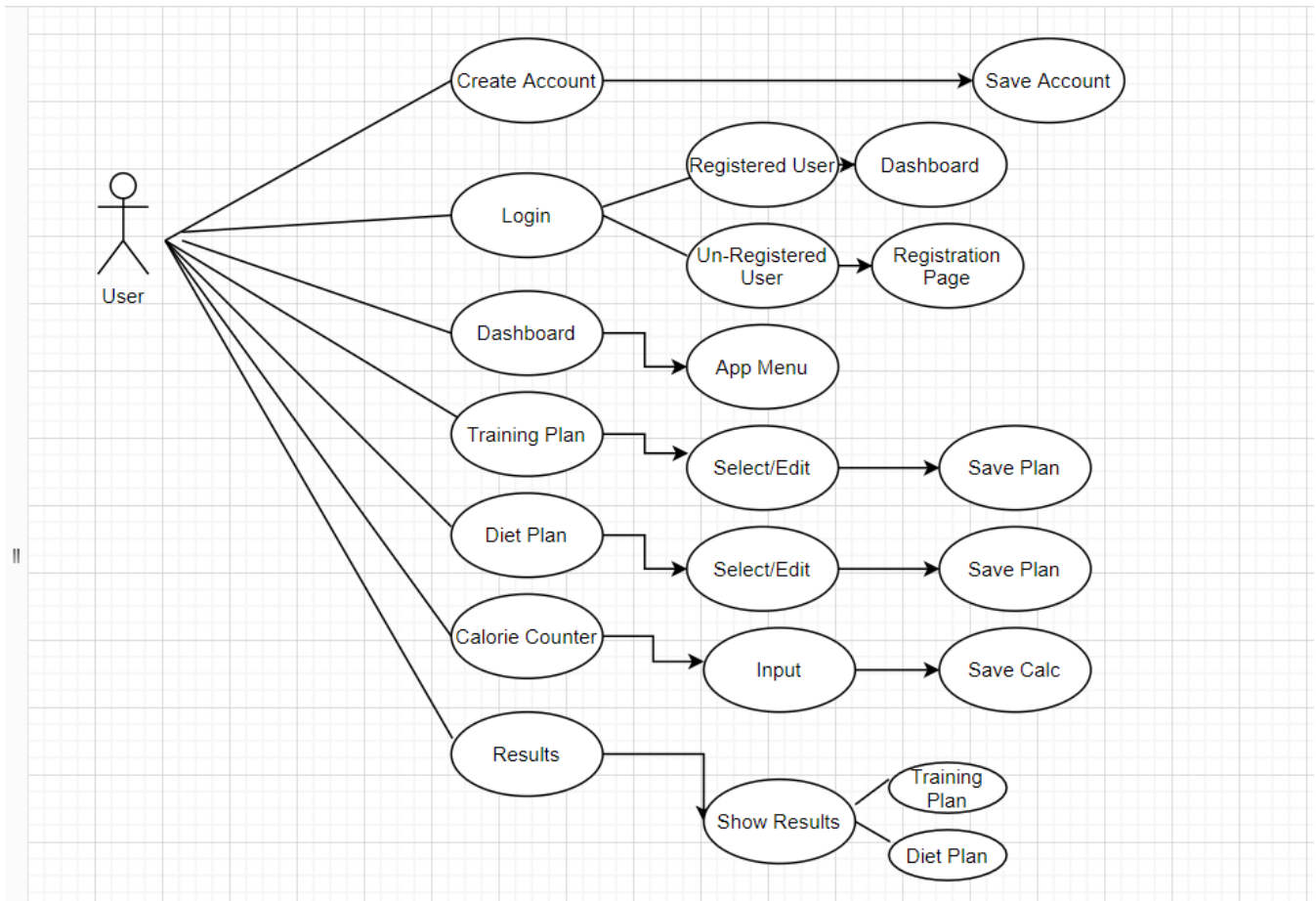
1. On Opening of dashboard users be able to interact with the Diet Plan button to access the GUI Diet Plan.
2. On loading of the GUI all saved programmes shall be loaded by the database.
3. All users shall have the ability to create and edit all their programmes from the Diet Plan GUI.
4. All users shall have the ability to save their programmes to be stored within the database.

Nutrition

1. On Opening of dashboard users shall be able to interact with the Nutrition button to access the GUI Nutrition.
2. The system shall load links to Instagram page and a website whereby it will give a detailed description of nutrition.

2.1.1.1. Use Case

Use Case Diagram



2.1.1.2. Requirement 1: User Registration/Login

A user starts the application and is presented with a GUI register screen where it allows users to register or if already registered can go straight to the Login GUI. The GUI presents text boxes where a user can input their details. The system can reject Registration if the fields are not correctly filled out. Once the user Registers they have the capability to log in and use the application.

2.1.1.3. Description & Priority

User Registration/Login will be the first image that comes up when the application is launched, this requirement is essential to the systems performance as the user will not be able to use the rest of the application if they do not input their login details or for new users not register. If a user does not register or fails to login there will be no use for the application as the app is ran off your personal details and each account is linked to one indivual.

Requirement 1: User Registration/Login

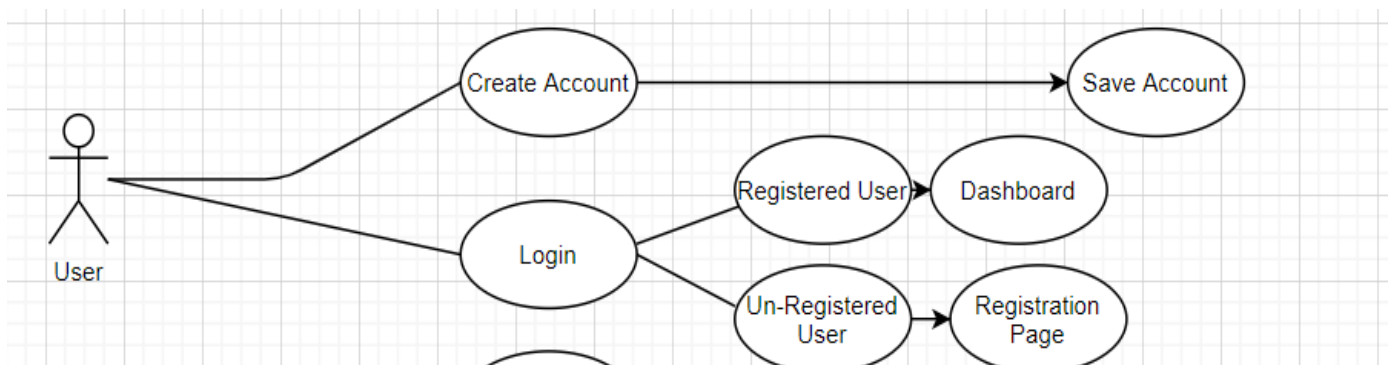
Scope

The Scope of this use case is to show how a user can register/Login to the application.

Description

This use case describes how a user would register for the application to use the application or if there an existing user they can login to use the application.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the Login/Register screen is loaded.

Activation

This use case starts when an <User> activates the register page or an already registered <user> logs in.

Main flow

1. The system identifies the Login/Register.
2. The <user> inputs their details.
3. The system then updates the database.
4. The <user> is then given access to the application.

Alternate flow

A1: <Register>

1. The system loads up the register page.
2. The <User> inputs their details.
3. The use case continues at position 3 of the main flow.

Exceptional flow

E1: <Login>

4. The system loads up the login page.
5. The <User> inputs their login details.
6. The use case continues at position 4 of the main flow.

Termination

The system presents the next state as the <User> has full access to the application.

Post condition

The system goes into a wait state when a user is not using the application for a period.

2.1.2. Data Requirements

User Data

1. The system shall store all user's login username and password securely and shall not share with anyone.
2. A user will have to register to login with a valid registration.

Programme Data

1. The system shall store all users saved and edited programmes and shall not share with anyone.
2. A user must create a programme to use the edit function.

2.1.3. User Requirements

Unregistered User

An unregistered user will not be able to gain access to use the applications features or interact with the application, they can only input into the text boxes. Once the user does not register, they have not agreed to any terms of service.

Registered User

Once completed the registration to the application users have full access to the application whereby, they can view programmes edit, and create programmes and also save programme. Once saved the system will save the programme for each individual account whereby it can be shared by a user to other members of the app. Once using the application users use the application at their own risk.

1. Requirement 2: Dashboard

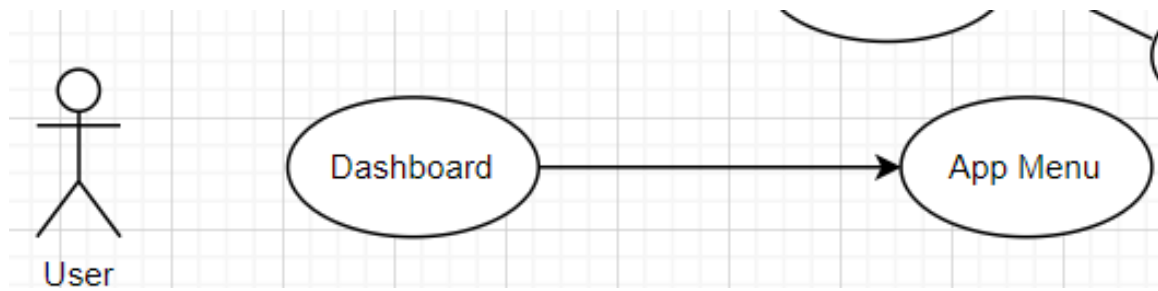
Scope

The Scope of this use case is to show how a user can gain accesses to the dashboard of the application once a successful login.

Description

This use case describes how a user would accesses the application to use the dashboard menus.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the User logs in successfully.

Activation

This use case starts when an <User> activates the login button and when is already a registered <user> .

Main flow

5. The system identifies the Login details.
6. The <user> inputs their details.
7. The system then updates the database.
8. The <user> is then given access to the dashboard menus of the application.

Alternate flow

A1: <Dashboard>

7. The system loads up the dashboard menus page.
8. The <User> can select from one of the buttons on the menus.
9. The use case continues as one of the options selected.

Exceptional flow

E1: <Dashboard>

10. The system loads up the dashboard page.
11. The <User> does not select a button from the menus.
12. The use case ends.

Termination

The system presents the next state as the <User> selects a button from the menus.

Post condition

The system goes into a wait state when a user is not using the application for a period.

2.1.4. Data Requirements

User Data

3. The system shall store all user's login username and password securely and shall not share with anyone.
4. A user will have to register to login to gain access of the dashboard.

Programme Data

3. The system shall store all users saved and edited programmes and shall not share with anyone.
4. A user must create a programme to use the edit function.

2. Requirement 3: Training Plan

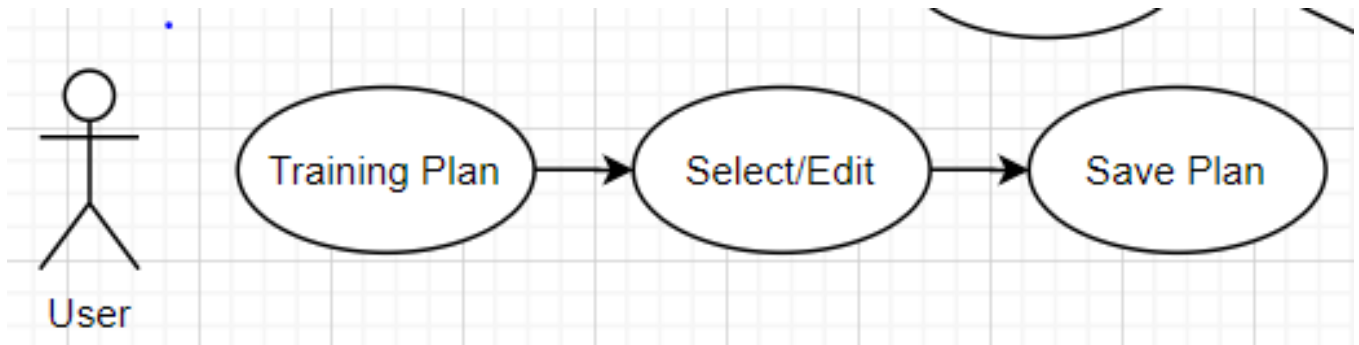
Scope

The Scope of this use case is to show how a user can gain accesses to the training plan feature of the application once a user has selected the button from the menus.

Description

This use case describes how a user would accesses the application to use the Training plan feature to add a workout plan.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the User selects the Training plan button.

Activation

This use case starts when an <User> activates the Training Plan button.

Main flow

9. The system identifies the button click.
10. The system loads up the Training plan form.
11. The user can then selected add and edit plans.
12. The user can go back to the dashboard by another button click.

Alternate flow

A1: <Training Plan>

13. The system loads up the dashboard menus page.
14. The <User> select from the Training Plan button on the menus.
15. The use case continue when the user selects a programme.

Exceptional flow

E1: <Training Plan>

16. The system loads up the Training Plan page.
17. The <User> does not select from the menus.
18. The use case ends.

Termination

The system presents the next state as the <User> selects from the menus.

Post condition

The system goes into a wait state when a user is not using the application for a period.

2.1.5. Data Requirements

User Data

5. The system shall store all user's selections securely and shall not share with anyone.
6. A user will have to selected Training plan button to gain access of the Training Plan form.

Programme Data

5. The system shall store all users saved and edited programmes and shall not share with anyone.
6. A user must create a programme to use the edit function.

3. Requirement 4: Diet Plan

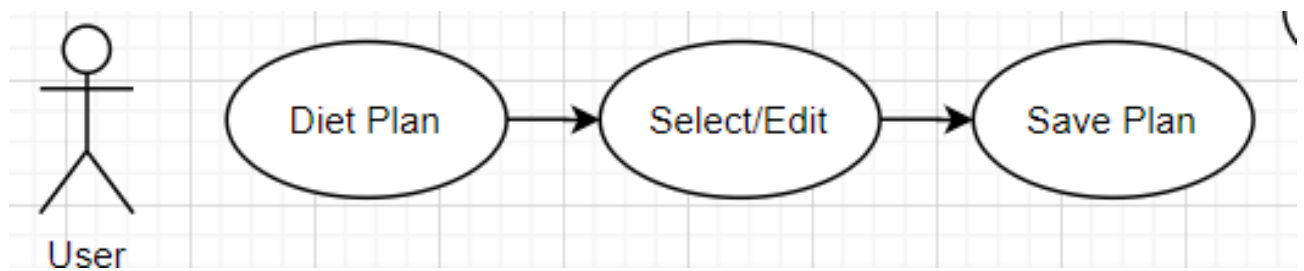
Scope

The Scope of this use case is to show how a user can gain accesses to the diet plan feature of the application once a user has selected the button from the menus.

Description

This use case describes how a user would accesses the application to use the diet plan feature to add a plan.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the User selects the Diet plan button.

Activation

This use case starts when an <User> activates the Diet Plan button.

Main flow

13. The system identifies the button click.
14. The system loads up the Diet plan form.
15. The user can then selected add and edit plans.
16. The user can go back to the dashboard by another button click.

Alternate flow

A1: <Diet Plan>

19. The system loads up the dashboard menus page.
20. The <User> select from the Diet Plan button on the menus.
21. The use case continues when the user selects a plan.

Exceptional flow

E1: <Diet Plan>

22. The system loads up the Diet Plan page.
23. The <User> does not select from the menus.
24. The use case ends.

Termination

The system presents the next state as the <User> selects from the menus.

Post condition

The system goes into a wait state when a user is not using the application for a period.

2.1.6. Data Requirements

User Data

7. The system shall store all user's selections securely and shall not share with anyone.
8. A user will have to selected Diet plan button to gain access of the Diet Plan form.

Programme Data

7. The system shall store all users saved and edited plans and shall not share with anyone.
8. A user must create a plan to use the edit function.

4. Requirement 5: Calorie Counter

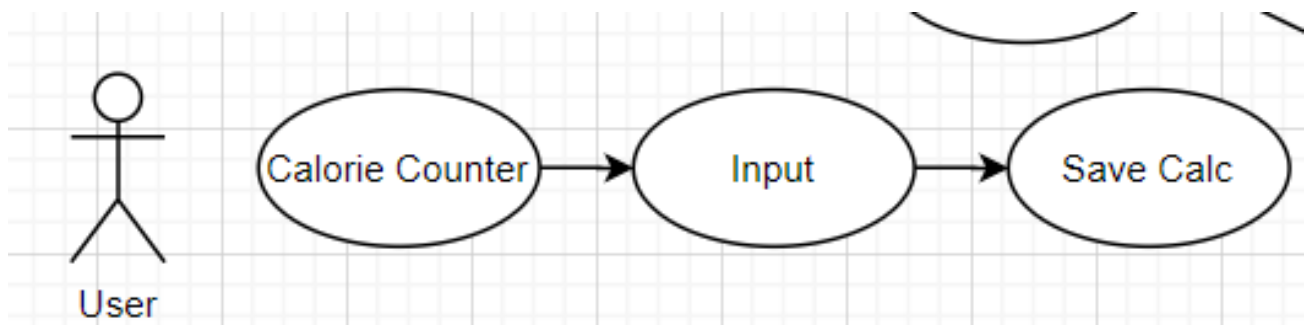
Scope

The Scope of this use case is to show how a user can gain accesses to the diet plan feature of the application once a user has selected the button from the menus.

Description

This use case describes how a user would accesses the application to use the diet plan feature to add, edit and select.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the User selects the Calorie Counter button.

Activation

This use case starts when an <User> activates the Calorie Counter button.

Main flow

17. The system identifies the button click.
18. The system loads up the Calorie Counter form.
19. The user can then input calories and will show recommended intake vs burned.
20. The user can go back to the dashboard by another button click.

Alternate flow

A1: < Calorie Counter>

25. The system loads up the dashboard menus page.
26. The <User> select from the calorie counter button on the menus.
27. The use case continues when the user inputs a numeric digit.

Exceptional flow

E1: < Calorie Counter >

28. The system loads up the Calorie Counter page.
29. The <User> does not select from the menus.
30. The use case ends.

Termination

The system presents the next state as the <User> selects from the menus.

Post condition

The system goes into a wait state when a user is not using the application for a period.

2.1.7. Data Requirements

User Data

9. The system shall store all user's inputs securely and shall not share with anyone.
10. A user will have to selected Calorie Counter button to gain access of the Calorie Counter form.

Programme Data

9. The system shall store all users input and results and shall not share with anyone.
10. A user must input a numeric digit the use edit function.

5. Requirement 6: Results

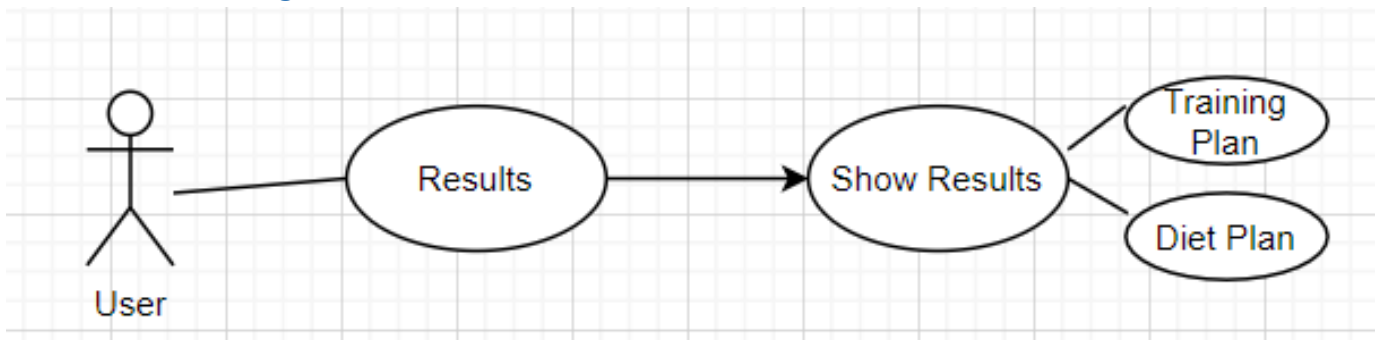
Scope

The Scope of this use case is to show how a user can gain accesses to the Results feature of the application once a user has selected the button from the menus.

Description

This use case describes how a user would accesses the application to use the feature to see the show results from Training Plan and Diet Plan.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the User selects the Results button.

Activation

This use case starts when an <User> activates the Results button.

Main flow

21. The system identifies the button click.
22. The system loads up the Results form.
23. The user is then shown results from TP and DP.
24. The user can go back to the dashboard by another button click.

Alternate flow

A1: < Results >

31. The system loads up the dashboard menus page.
32. The <User> select from the results button on the menus.
33. The use case ends.

Exceptional flow

E1: < Results >

34. The system loads up the Calorie Counter page.
35. The <User> does not select from the menus.
36. The use case ends.

Termination

The system presents the next state as the <User> selects from the menus.

Post condition

The system goes into a wait state when a user is not using the application for a period.

6. Requirement 7: About us/Nutrition

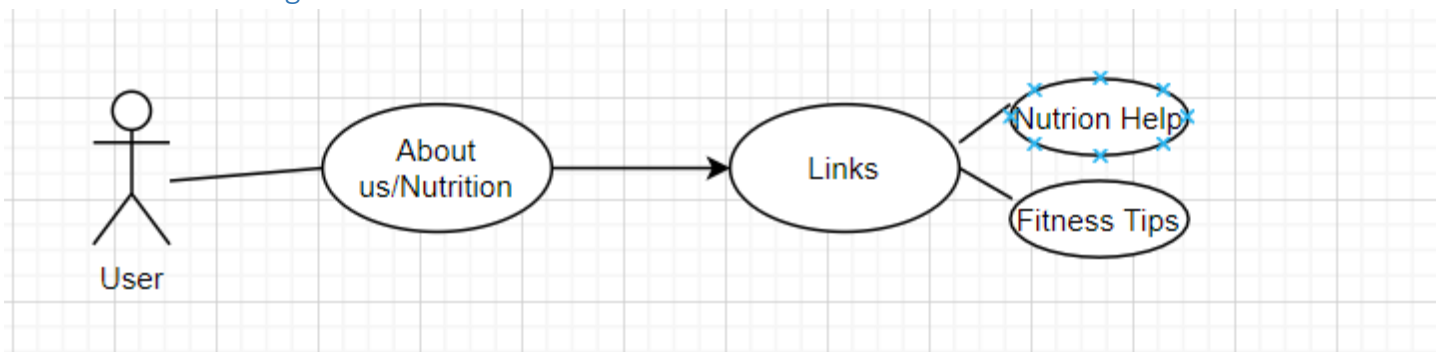
Scope

The Scope of this use case is to show how a user can gain accesses to the About us/Nutrition feature of the application once a user has selected the button from the menus.

Description

This use case describes how a user would accesses the application to use the feature for help and tips.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode when the User selects the About/Nutrition us button.

Activation

This use case starts when an <User> activates the About us/Nutrition button.

Main flow

25. The system identifies the button click.
26. The system loads up the Abouts us/Nutrition form.
27. The user is then shown About the brand and also links for fitness tips and diet help.
28. The user can go back to the dashboard by another button click.

Alternate flow

A1: < About us/Nutrition >

37. The system loads up the dashboard menus page.

38. The <User> select from the about us/nutrition button on the menus.

39. The use case is then continued if the user selects the links which will be linked to helpful pages.

Exceptional flow

E1: < About us/Nutrition >

40. The system loads up the About us/Nutrition page.

41. The <User> does not select from the links.

42. The use case ends.

Termination

The system presents the next state as the <User> selects from the menus.

Post condition

The system goes into a wait state when a user is not using the application for a period.

2.1.8. Environmental Requirements

The application aims to work on all android smartphones and tablets as of 2019 models and will be coded mainly using Visuals studios C#, the information of both the login and registration will be stored in MS access database where also the workouts and saved programmes will be stored.

2.1.9. Usability Requirements

I used an Ms access connection within my code to check the usability of both user login and registration. I also tested that the application was runnable on visual studio code free of errors. I done some usability testing on when a user does not input that an error occurs and the user I brought back to the text box to input straight after the error.

I made sure that all text within the application from user login, registration, dashboard, and all forms was readable and also the user inputs fit in the textbox.

I made sure that all dimensions within the application were the same of all the forms and also the colour used was the same through the application.

I designed the application so that the registration is always the first form and also that a user must be a useful registrant in order to login and access the dashboard features.

Another feature for which I added was a user must have the same password for both password inputs in the registration page where they must confirm their password and also can show their password if needed.

In the Diet Plan and Training form a user must input characters before adding a plan if not a message box will pop up to notify them to input into the fields and will redirect them back to the first text box on the form.

With using the icons I had to make sure that all icons fit in the buttons and that there were also readable by the user.

2.2. Design & Architecture

The application will be designed to cater for smartphones and tablets, the system I will be using for my Login and register is Visual studio 2019 and coding will be done in C# using Windows forms. I will also be using other components within Visual studio to complete my Project. The main algorithm within the project is by where a user's information and training plan will be stored safely in the data base and will be confidential to each account. The application designed changed as the project went on as the computer device I am using limited myself to adding certain aspects of the desired project idea.

2.3. Implementation

The key functional requirements of the Gym program will be described in this part, along with code snippets. Some of the code snippets displayed show of how I connected my visual studio forms to my database.

Register New User

The code snippet below describes a user registering for the application and how to information is connected to a to store the credentials within the data base.

```
public OleDbConnection con = new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=db_users.mdb");
public OleDbCommand cmd = new OleDbCommand();
public OleDbDataAdapter da = new OleDbDataAdapter();

1 reference
private void button1_Click(object sender, EventArgs e)
{
    if(txtUsername.Text == "" && txtPassword.Text == "" && txtComPassword.Text == "")
    {
        MessageBox.Show("Username and Password fields are empty", "Registration Failed", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else if(txtPassword.Text == txtComPassword.Text)
    {
        con.Open();
        string register = "INSERT INTO tbl_users VALUES('" + txtUsername.Text + "','" + txtPassword.Text + "')";
        cmd = new OleDbCommand(register, con);
        cmd.ExecuteNonQuery();
        con.Close();

        txtUsername.Text = "";
        txtPassword.Text = "";
        txtComPassword.Text = "";

        MessageBox.Show("Your Account has been Created Successfully", "Registration Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        MessageBox.Show("Passwords does not match, Please Re-Enter", "Registration Failed", MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtPassword.Text = "";
        txtComPassword.Text = "";
        txtPassword.Focus();
    }
}
```

User Login 1

This code snippet will show the data base connection with the login credentials and show an if statement whereby the system will throw an error if the details are invalid on Login.

```
namespace GymAppProject
{
    3 references
    public partial class frmLogin : Form
    {
        1 reference
        public frmLogin()
        {
            InitializeComponent();

            public OleDbConnection con = new OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=db_users.mdb");
            public OleDbCommand cmd = new OleDbCommand();
            public OleDbDataAdapter da = new OleDbDataAdapter();

            1 reference
            private void button1_Click(object sender, EventArgs e)
            {
                con.Open();
                string login = "SELECT * FROM tbl_users WHERE username= '"+txtusername.Text+"' and password='"+txtpassword.Text+"'";
                cmd = new OleDbCommand(login, con);
                OleDbDataReader dr = cmd.ExecuteReader();

                if(dr.Read() == true)
                {
                    new dashboard().Show();
                    this.Hide();
                }
                else
                {
                    MessageBox.Show("Invalid Username or Password, Please Try Again", "Login Failed", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    txtusername.Text = "";
                    txtpassword.Text = "";
                    txtusername.Focus();
                }
            }
        }
    }
}
```

User Login 2

This code snippet will show how the registration page is linked to the login whereby if a user is not registered, they will not be able to have full use of the application and also if the register successful and also if the passwords and username are correct.

```
1 reference
private void button2_Click(object sender, EventArgs e)
{
    txtusername.Text = "";
    txtpassword.Text = "";
    txtusername.Focus();
}

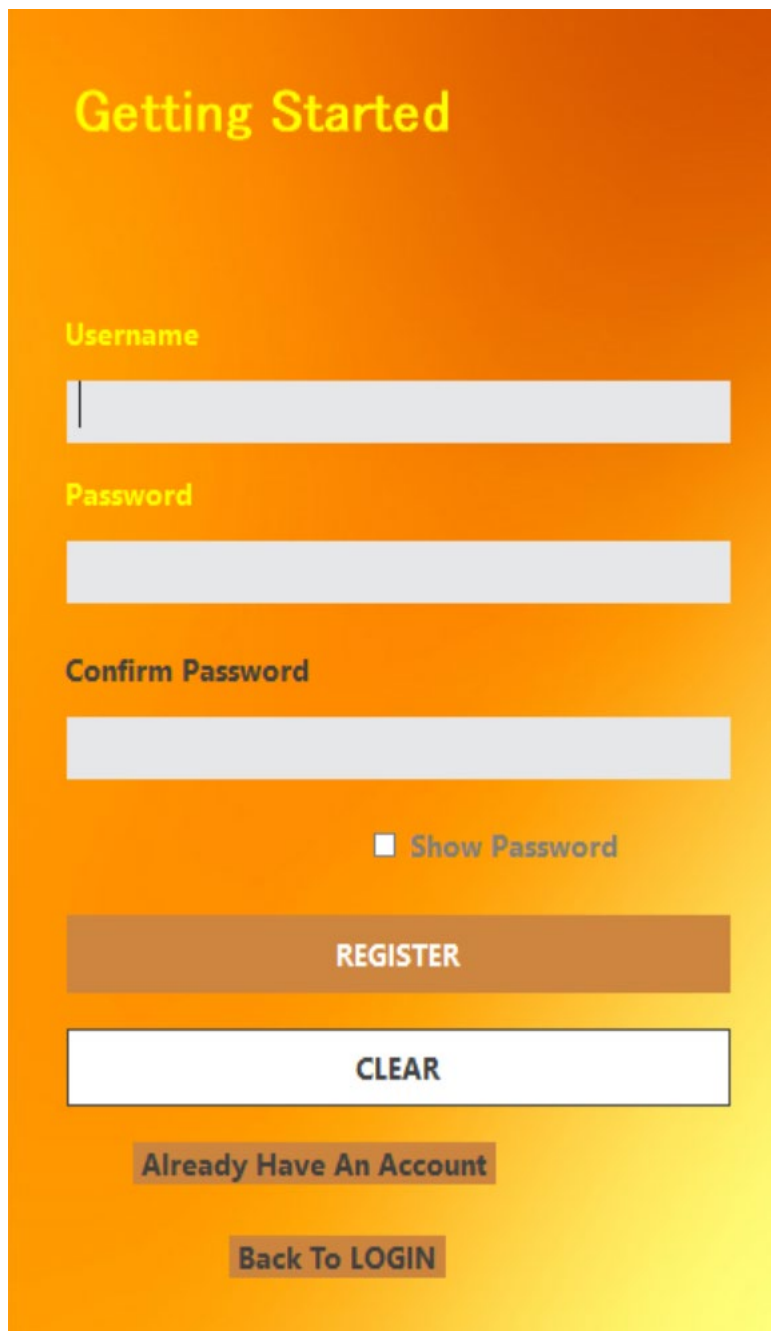
1 reference
private void checkboxShowPassword_CheckedChanged(object sender, EventArgs e)
{
    if (checkboxShowPassword.Checked)
    {
        txtpassword.PasswordChar = '\0';
    }
    else
    {
        txtpassword.PasswordChar = '*';
    }
}

1 reference
private void label6_Click(object sender, EventArgs e)
{
    new frmRegister().Show();
    this.Hide();
}
}
```

2.4. Graphical User Interface (GUI)

Registration Page

The Registration screen allows all users to create a username and password and all users must confirm their password in order to gain access to the login. If already registered a user may click on the back to login.



The registration page features a vertical orange-to-yellow gradient background. At the top, the text "Getting Started" is displayed in a bold, yellow, sans-serif font. Below this, the form fields are labeled in a smaller, yellow, sans-serif font: "Username", "Password", and "Confirm Password". Each label is followed by a light gray rectangular input field. A checkbox with a small white square icon is positioned to the left of the text "Show Password". Below the input fields, there are three buttons: a solid brown button with the word "REGISTER" in white, a white button with a black border and the word "CLEAR" in black, and a solid brown button with the text "Already Have An Account" in white. At the bottom, there is a solid brown button with the text "Back To LOGIN" in white.

Getting Started

Username

Password

Confirm Password

☐ Show Password

REGISTER

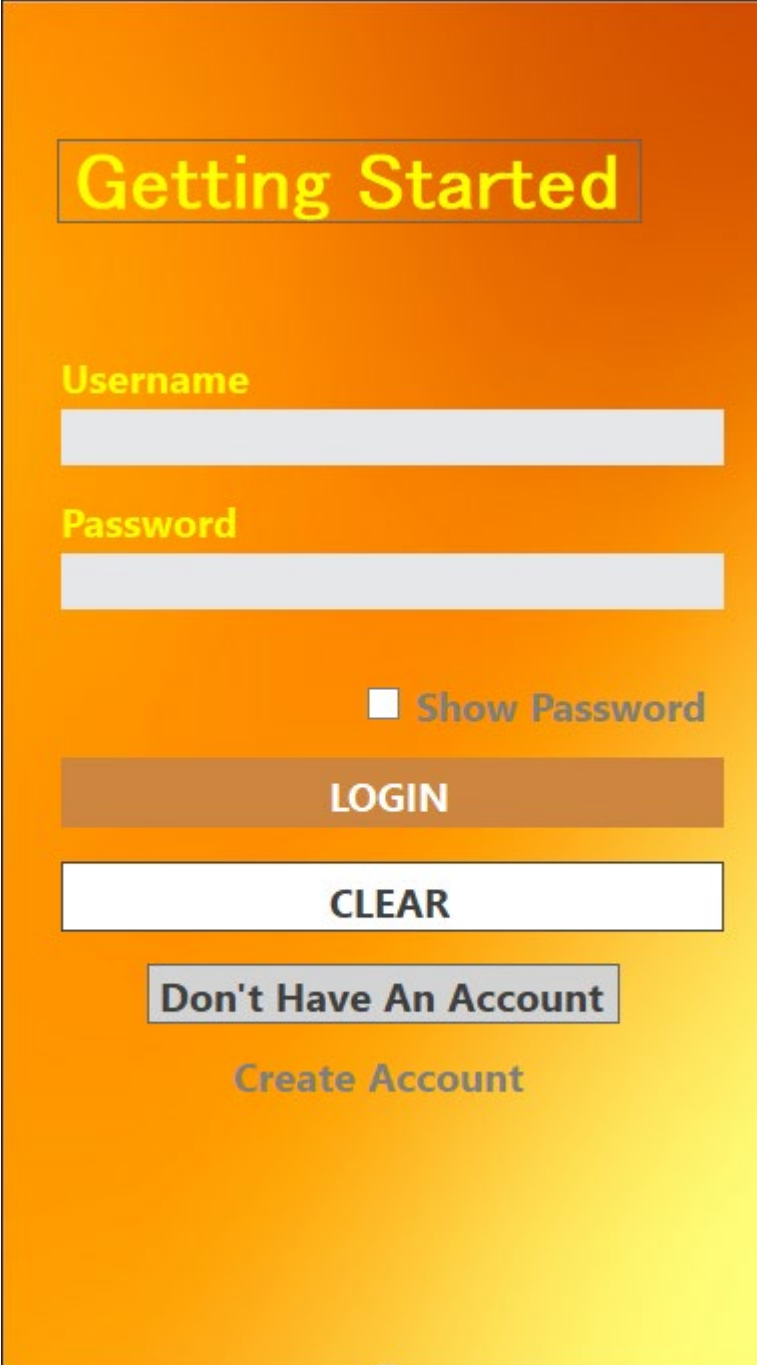
CLEAR

Already Have An Account

Back To LOGIN

Login Screen

The login screen will be loaded on clicking on the back to login on the registration page, once a user is registered then can login by using their details for which they used to complete the registration. I have implemented a show password button as some users of the application might have to see their password while inputting in the text boxes. Also with the ability for users to create new accounts.



The login screen features a vertical orange-to-yellow gradient background. At the top, the text "Getting Started" is displayed in yellow within a thin orange-bordered box. Below this, the label "Username" is in yellow, followed by a light gray text input field. The label "Password" is also in yellow, followed by another light gray text input field. To the right of the password field is a checkbox with the label "Show Password" in gray. Below the input fields are two buttons: a brown "LOGIN" button and a white "CLEAR" button with a gray border. At the bottom, there is a gray button labeled "Don't Have An Account" and the text "Create Account" in gray below it.

Getting Started

Username

Password

☐ Show Password

LOGIN

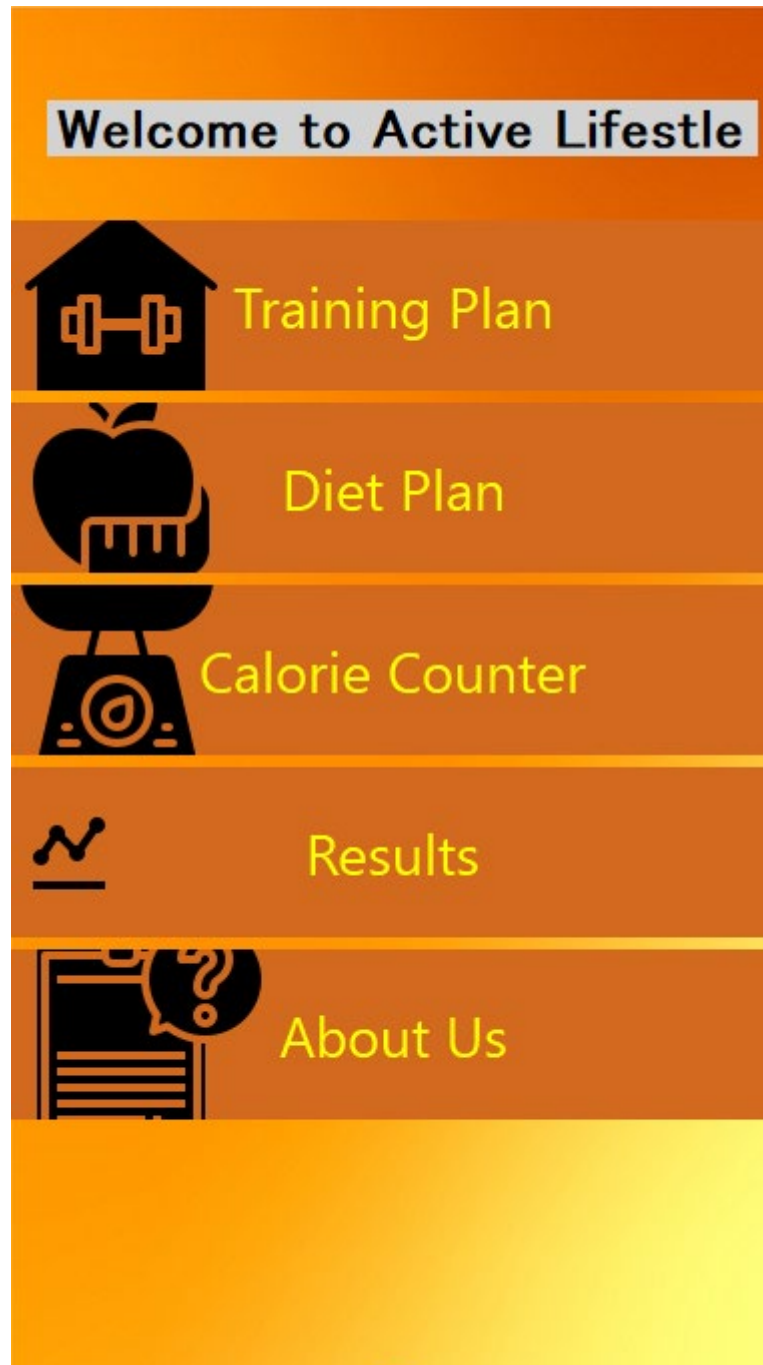
CLEAR

Don't Have An Account

Create Account

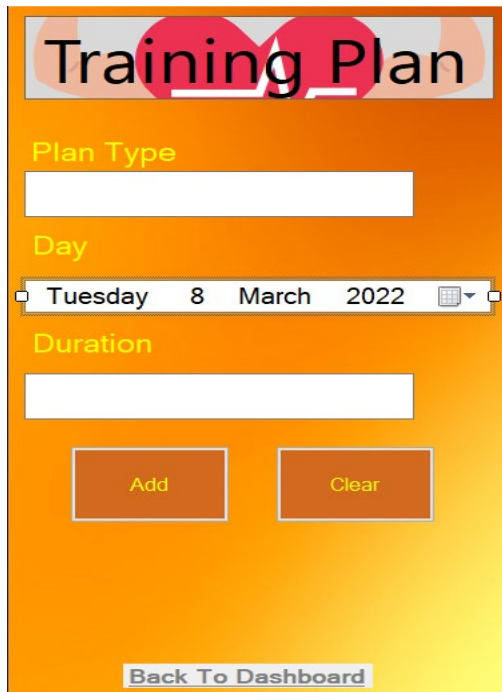
Dashboard

Once the user has successfully logged in and has been recognized by the data base as a registered user the loading of the dashboard will commence. The user dashboard includes 5 buttons for which the user can interact with. The buttons once clicked sends the user to the desired form for which button click. For example, if the user clicks the training plan button the training plan form will load for the user. I changed the styling of the GUI to give it more attractive look for users.

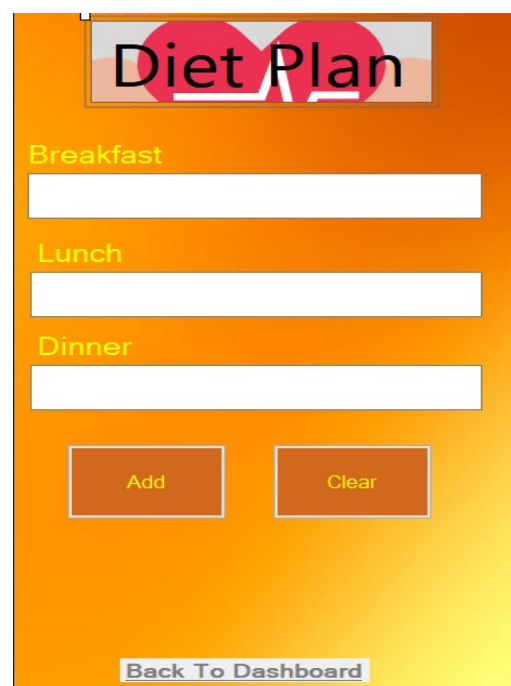


Training Plan/Diet Plan

Once the user has clicked on the training plan button on the dashboard, the training plan gui will load up letting the user input there training plan type, the day and the duration of the training. Also, the same process happens with the diet plan button on the dashboard. Here a user can add and also clear their inputs and also can go back to the dashboard if the link is clicked.



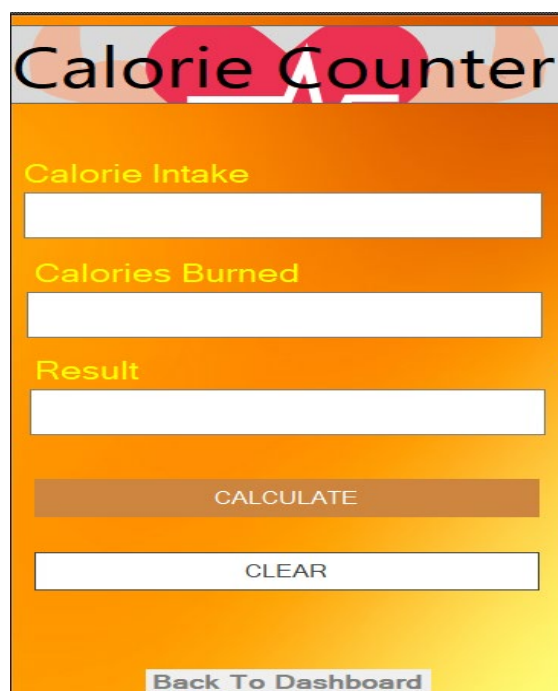
The Training Plan GUI has an orange-to-yellow gradient background. At the top, the title "Training Plan" is in a grey box. Below it, the label "Plan Type" is in yellow, followed by a white input field. The label "Day" is in yellow, followed by a date picker showing "Tuesday 8 March 2022". The label "Duration" is in yellow, followed by a white input field. At the bottom, there are two orange buttons labeled "Add" and "Clear". A "Back To Dashboard" link is at the very bottom.



The Diet Plan GUI has an orange-to-yellow gradient background. At the top, the title "Diet Plan" is in a grey box. Below it, the label "Breakfast" is in yellow, followed by a white input field. The label "Lunch" is in yellow, followed by a white input field. The label "Dinner" is in yellow, followed by a white input field. At the bottom, there are two orange buttons labeled "Add" and "Clear". A "Back To Dashboard" link is at the very bottom.

Calories Counter GUI

Once the user has interacted with the Calorie Counter button, the Calorie Counter GUI is loaded which allows the user to input there calories taken in and also the calories burned and once the user has interacted with the calculate button a user can see there results.



The Calorie Counter GUI has an orange-to-yellow gradient background. At the top, the title "Calorie Counter" is in a grey box. Below it, the label "Calorie Intake" is in yellow, followed by a white input field. The label "Calories Burned" is in yellow, followed by a white input field. The label "Result" is in yellow, followed by a white input field. Below the input fields are two buttons: a brown "CALCULATE" button and a white "CLEAR" button. A "Back To Dashboard" link is at the very bottom.


About us GUI

Once the user has interacted with the About us button, the About us GUI is loaded which describes the application and also gives an email for which is to deal with any questions and also includes 3 links to a helpful Instagram page, YouTube videos on workouts and also nutrition and training advice website. Once a user interacts with any of the 3 buttons it will bring them to the desired location for which icon they click. The first icon is for Instagram which will link to an Instagram page for active lifestyle, the next icon is a youtube icon which links to youtube channel for help for training and the next icon is for nutrition help.



Results GUI.

When the user interacts with the result button the GUI for results appears which shows 2 check box and a print button. A feature which causes a lot of stress and time in the development of the project for which I could not get fully working as when I clicked the checkbox and tried to print the results would not show as appeared on the data base. Therefore, I had to input images to show what it should have looked like and on the print results, I finally concluded that this page was causing more harm to the application when running that I had to make it static. I am really annoyed with myself as I did not fully evaluate the form but caused too much time and was causing errors within in the application. This feature is one that in the future I hope to complete, if I was doing this project again, I would give myself more time focusing on the right practises and the data connection and also a new device as mine is old and wasn't compatible with the bit size of visual studios to directly connect with Ms Accesses.



The screenshot shows a GUI titled "Results" with an orange-to-yellow gradient background. At the top, there is a header with the word "Results" in a large, bold, black font. Below the header, there are two checkboxes: "Show Diet Plan" and "Show Training Plan". Below these checkboxes, there are two data tables. The first table, titled "tbl_trainingplan", has columns for ID, Plan, Day, and Duration. It contains one row with the value "1 Full Body Weights" for ID, "28/02/2022 40 mins" for Day, and "40 mins" for Duration. The second table, titled "tbl_dietplan", has columns for ID, Breakfast, Lunch, and Dinner. It contains one row with the value "1 Porridge" for ID, "Pasta" for Lunch, and "Steak and Veg" for Dinner. At the bottom of the GUI, there is a "Print" button and a "Back To Dashboard" button.

ID	Plan	Day	Duration
1	Full Body Weights	28/02/2022	40 mins

ID	Breakfast	Lunch	Dinner
1	Porridge	Pasta	Steak and Veg

Print

Back To Dashboard

Testing

<https://github.com/x18510469BK/GymAppProject.git>

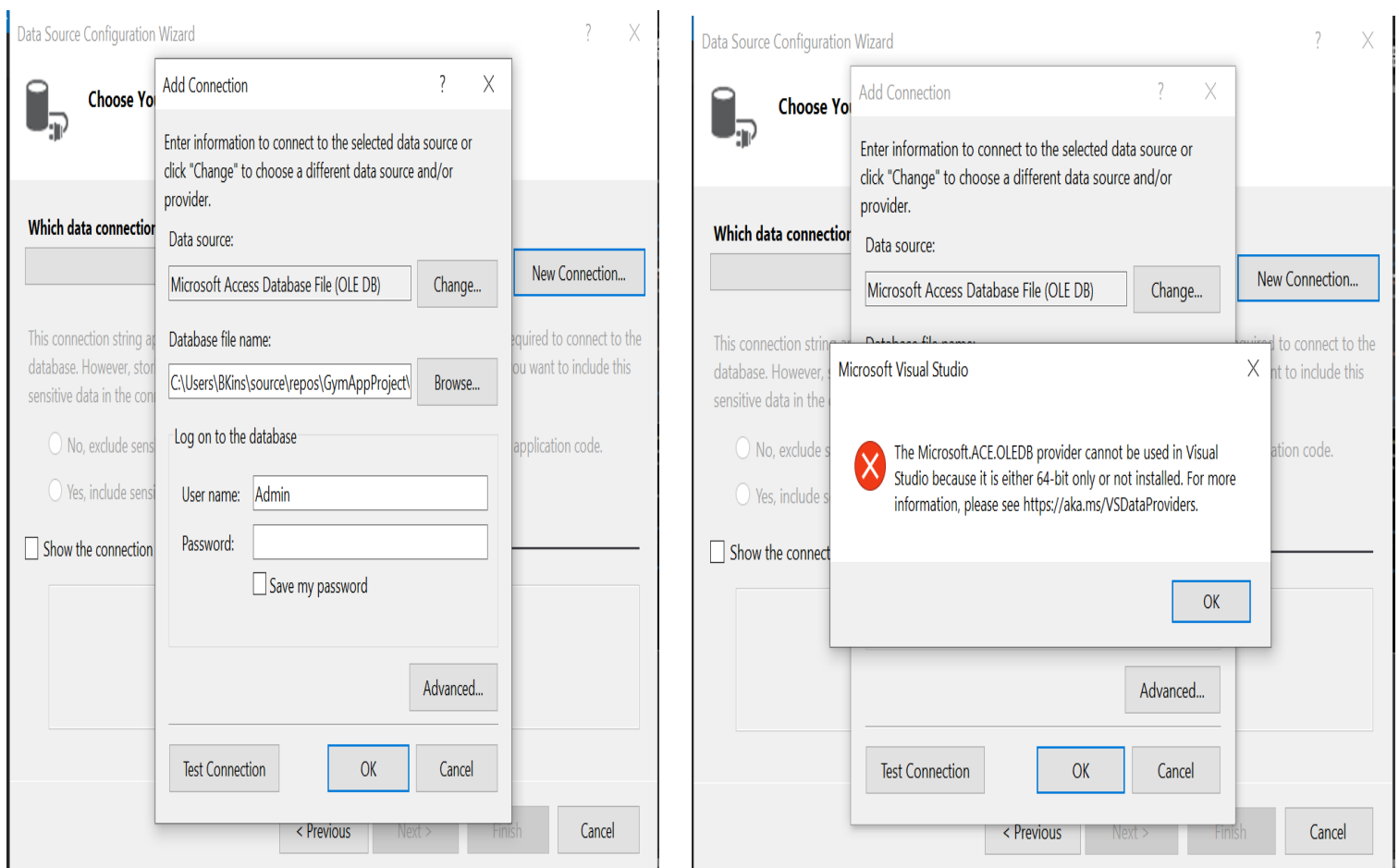
2.5. Testing

Testing was carried out by myself personally as I have been studying remotely, I done a quick survey both in my house and using social media I created a poll for which I gained more

knowledge and information about the numbers that avoid the gym because of its complicity and a lot of people with no confidence. The survey was carried out over a day and will be carried out again on the progression of my project.

I used Ms excel to help with my Unit testing of the application and its various features this helped me with seeing the progress of each and the workings of the application.

Testing on the database connection took some time as my computer was having problems with the Ms access connection also, I tested the connection in visual studios itself and came across problems.



2.6. Evaluation

Evaluation of my application, as the results from my first testing were very biased towards one side as most of my family and close friends show an interest in my application idea. I will carry out many tests once my project is in the progression stage that its efficient to carry out

surveys again. I used Unit testing as a way of evaluation of my application which was very helpful in the creation and design of my application as it kept me on track and could also check off for which parts were fully completed and which parts needed tweaking.

3.0 Conclusions

In conclusion to the testing which took place I have gained data for which the application can be improved and some functionality for which I should take better research into be implementing in my project. The idea of my project and the market for which I am targeting is a strength to my project, but the application can only cater for exercise workouts and does not let someone plan there week around sporting activities which was recommended to me and I will try implement this aspect in my further progression of the project. In conclusion to completing my application I could have done a lot of planning better and also that my device for which I was using was extremely slow to download certain functionalities which made my project be restricted as I had to simplify it and get the application working instead of having an application with various features which did not work. I leaned a lot about why its so important to do research and also have a structured plan in the project for which I had taught I had sorted but this was not the case, coming into so many problems and errors was a stress for me when trying to develop the perfect application, I felt that learning remotely for nearly 2 a half years of the 4 years had been a major disadvantage for my learning of the various practises and also web development for which we only had once over the 4 year period.

4.0 Further Development or Research

As reading further development I would have liked for my application to have the functionality to be improved as the users of the application increase, I would scope to look for my materials and tools for which could be implemented in the development of my application. After my project I would also like to make my application compatible with all IOS and android products as the market would be higher and the users of the application. I will hope to be using my application for my own personal training and be able to help my friends and family through the application.

5.0 References

<https://docs.microsoft.com/en-us/visualstudio/deployment>

<https://docs.microsoft.com/en-us/dotnet/desktop/winforms/forms/how-to-add>

<https://docs.microsoft.com/en-us/visualstudio/data-tools/connect-to-data-in-an-access-database-windows-forms>

<https://stackoverflow.com/questions/41459165/visual-basic-password-generator>

<https://www.microsoft.com/en-ie/microsoft-365/p/access>

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/>

<https://moodle2020.ncirl.ie/course/AdvancedInternetTechnologies>

<https://docs.microsoft.com/en-us/visualstudio/extensibility/ux-guidelines/colors-and-styling-for-visual-studio>

<https://docs.microsoft.com/en-us/visualstudio/data-tools/connect-to-data-in-an-access-database-windows-forms?view=vs-2022>

6.0 Appendices

6.1. Project Proposal

1.0 Objectives

My project will be a Fitness Application based around the main topics that people have with Fitness. For example, many people would have a fear off attending a gym for the fact they would be unsure on what to do and also people that attend the gym have a problem with the time spent and also remembering which workout routine to do. This application will be hopefully integrated with Polar API to monitor heart rate duration of training. A user can create an account (Name, Age, Height, Weight, Nationality, Athlete) which will then allow them to share their routines with other users and create a structured plan alongside diet. The User will also be able to tailor their training for a specific time lapse or can also add members to their plan so if a user is training with a partner, then can both be on the same plan. At the end of each session the user will get back a vary of results including (Heart rate, Calories burned, Calories In-Calories burned, Time of the session). The user can also set days for rest days and can keep track of their food on that day. At the end of each Users Week the user will get a summary of the week training. I will hope to add in a camera function whereby they can take pictures to see the progress. All the Users details will be stored on a data base.

There were many changes made to both functionality and also the design of the application as my device was very slow to operate with the various heart rate and google features in my application. Also, I had a lot of problems when adding members to the application as again my device would only allow for the use of Ms access as my SQL Server port had been changed on my device. I tried and tested my application with these features, but it would not run without throwing errors. This has caused a lot of stress when trying to complete my project efficiently and on time.

2.0 Background

I choose this project as it relates to my everyday life and many of the people, I surround myself with life, I am always having these problems in the gym and when I train outdoors for which I don't know if I'm doing too much or too little. The application also helps implement your

daily diet which would help hugely alongside training to live a healthy life. The application caters for all levels and aspects of training to help training easy and not be the thing that people give out about or is made to be a task.

3.0 State of the Art

There is a few similar applications I have both used and re-searched but none exactly like mine. There are many apps whereby they track your run, track your calories, design your own workout, track your heartbeat, but my app allows users to integrate all these in one and will have the ability to also count calories. A lot of these apps are hard to navigate and understand my application will have sections for beginners right up to elite athletes. The app is designed to show how easy it is and how much little time is needed for a healthy life.

4.0 Technical Approach

Development will be a long process as I have many commands in the application for the User and getting my application to run with Polar API will be a task. I will break my project down in to 3 sections. (App Creation, Data base, Server). I will hope to have my app developed and linked with the Polar API before the second semester. Then I can start working on Linking to a data base to store user details and the creation of users being able to share their workouts and plans with each other. There were many different technical approaches used after the midpoint submission as I tried many other ways to create the application through different platforms and using different data base servers, this took me a lot of time and time was running against me, so I had to settle for using VS and Ms Access as it was the only applications for which did not slow down my device to be able to complete the project.

5.0 Technical Details

For this project I will using visuals studios for the development and design of the application alongside C# for the implementation of the API. I have to approach this project by implementing everything I have suggested whereby a user must be able to create and account and obtain all details to save in the data base and also to create there workout for the week

and also tailor there diet to there week, another important aspect is by the end of the week a user must obtain a weeks results to show progress or else that there is no progress.

6.0 Special Resources Required

Polar API, Google Maps API, Ms Access, Xamp, Android Emulator, YouTube Link, Instagram Link, Web Link.

7.0 Project Plan

Steps:

- Creation of Login Page/Registration.
- Design of Home Page.
- Design of weekly calendar.
- Design of Workout/ Diet Sheet integrated with Day.
- Linking Polar and Google Maps API to Application.
- Testing Of Application.
- Link To Data Base.
- Host Application.

Above is a simple summary of the steps that will be taken in order for the development of my project. I will start by using C# to create both the Login and Registration page. For the design of the weekly calendar, I will be using an API for this but haven't chosen yet as it is not an important step. For the design of the workout/diet sheet I will be using android studios. I will then have to use API to link the google Maps and Polar to my application. I will then have my data base linked whereby all user data for their own account is stored in history to track progress and go back and see for comparison of types of training. I haven't chosen of yet where I will host the application this will be chosen closer to the time.

8.0 Testing

The testing Process is the best part for me as I will be one of the end users using it for the functionality of my training for one week, I will also test my application against a competing application to show the difference between both apps. The testing process will also be used

by other members of the gym for which vary in age to show how easy it is to navigate and use.

6.2. Reflective Journals

September 2021

Getting back to college for myself was very hard and stressful as I had lost my Moodle account and also student email, as I could not gain contact with the college in regards my account I had to try get in contact with my peers to contact the college, I missed the first 3 weeks of the software project lectures and was really stressed about my situation in the college which was out of my hands.

October 2021

Finally, after months on end I was back on Moodle and was able to attend the teams meetings, I was told by a peer to get in contact with Frances in regarding the software project, she helped me straight away and told me that I have to make a proposal for my idea for which needed to be upload. I Started by choosing for which application I would use for design and also the deployment of the application.

November 2021

I concluded that coding in C# in visual studios had everything for which my application needed and that I would use Xampp and MS Access as my data base. For the development of the application and to run I have chosen to use Android Emulator an application whereby the android applications are running and tested. Before any coding or design was done, I made use cases for which I could see my project and how It was going to work. I started the design and started bits of coding as the month was packed with other modules CAs.

December 2021

I had completed my other modules CA and went back to finish of the tech report and Login form and make a database connection with Xampp. I had a lot of problems with this the username and password were not saving to my database. I concluded that Xampp was running too slowly on my device, and it was running on ports for which I had been using for other projects, so I decided to

change to Ms access after reading a few articles I discovered that MS access worked better for my project.

January 2022

After I had uploaded all terminal-based assessments for each of the modules, I could finally start the design of the dashboard before commencing the next semester. I was under a lot of pressure with the terminal-based assessments due to the time period of Christmas and also working a lot of hours in work, I thankfully uploaded them on time and got a small start on the dashboard and tried again to get my SQL working on my laptop but failed again and then it was time for the new semester.

February 2022

At the start of this month I had planned to start the integration of the API and finish the work on the dashboard but with the new modules and one being a data analysis module I had to learn new applications so it took up most of my time in the month , it was only towards later in the month I tried implementing the APIs and had many errors as my application would not run correctly with the APIs. Very stressed and with small knowledge of app development it was very difficult for me to implement the heart rate API in the time I had. I had done more small work on my showcase profile and had that completed which was another box ticked of the FYP. In this month we were also given CAs for our other modules.

March 2022

March was probably the most challenging month of my whole time in NCI, I was working a part time job and also helping my dad out with his company while trying to get the time to do CAs and my final bits of work on my project and complete my showcase profile for the FYP. I struggled in March to get any coding done as my priority was competing my CAs on time and with quality. I completed two of my CAs and gave myself time to again try the API and see if I could get it working, again I failed to get it working so I took it upon myself to not add the attribute into my project as it was taken to much time. As my SQL was suggested to me to use for the Database, I tried to fix the port on my computer but it still wouldn't connect so MS Access was the only option. I got small bits of time around work and college to work on the design of the dashboard and login and changed a few small features for the Training plan and diet plan.

April 2022

April became a very stressful period for myself as I concluded that Ms Access and Visual studio were the only applications for which my device would operate without been slow, I had to change many of my ideas and also could not incorporate both APIs which really annoyed me. Along with CAs and

TABAs due for this month I had prioritised getting my TABAs a CAs done before starting more work on the project when I had the time to do so. I had to change a lot of functionality within my app and also came across more errors when trying to print the results. Endless amount of searching online and no solutions it was on my own behalf that it was better to have a working application than one with many features and cool things that did not work. I got my training plan and diet plan linked to my data base and also created my abouts us form. Towards the end of April I got as much as I could done on both the document, showcase profile and the final bits of code in my application and also set up my Git Hub repository for my application. It was only in April that I found myself that I did not manage the time well with work and college and also the planning of the application was very vague as I came into the first semester 5 weeks late and had 1 week to do a project proposal and had to stick with my idea.

May 2022

In this month I had began the month set all the foundations for the project and actually got to see full representations of how my app will work, I linked the diet and training plan features to my Ms access data base but had some trouble printing the results from the data base to the results GUI. I had completed my technical report document and began trying to host my application again and also tried yet again to connect with MS Access directly, I honestly don't know how my computer device got through the project as for being so slow and going dead constantly while working on the project. I had fully completed my showcase profile and document so they were ready for upload and am I really hoping that I can pass this module as it has caused myself endless amount of stress and time but I really enjoyed learning things for which I did not learn on my course that I will hope to bring on for my future job if I pass. Contracting the virus and having problems with my eyes has also effected my work over the final days as I have been sick for over 10 days now and have sensitive vision so it has been very hard to get anything more done in the final days which has really been annoying for me both mentally and physically.

GitHub Repo

<https://github.com/x18510469BK/GymAppProject.git>

The screenshot shows the GitHub interface for a repository named 'GymAppProject' owned by 'x18510469BK'. The repository is marked as 'Private'. The top navigation bar includes links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the repository name, there are tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected, showing the 'master' branch with 1 branch and 0 tags. A commit history table is displayed, showing a commit 'a1756ef' from 6 hours ago with 2 commits. The commit message is 'BKins and BKins Add project files.' The table lists the files added: 'GymAppProject', '.gitattributes', '.gitignore', and 'GymAppProject.sln'. At the bottom, there is a prompt to 'Add a README with an overview of your project.' and a button to 'Add a README'.

File	Commit Message	Time
GymAppProject	Add project files.	6 hours ago
.gitattributes	Add .gitignore and .gitattributes.	6 hours ago
.gitignore	Add .gitignore and .gitattributes.	6 hours ago
GymAppProject.sln	Add project files.	6 hours ago