**MyCal**

**Analysis**

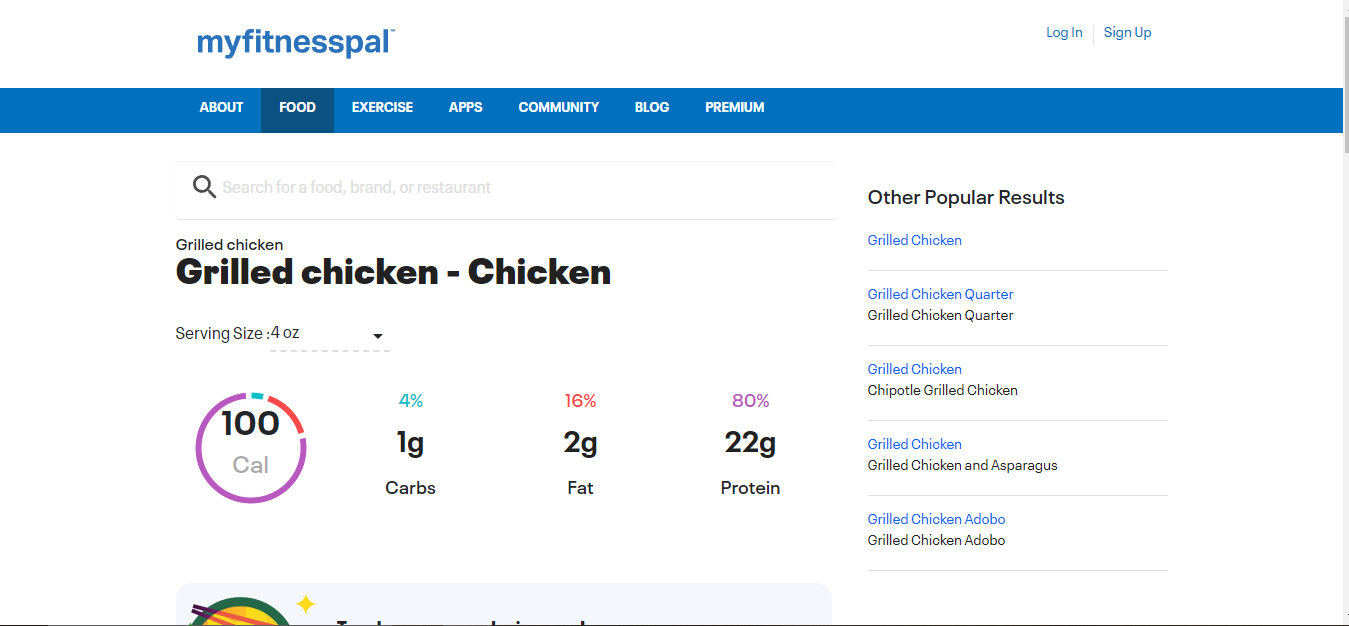
**Describe the problem:** Many people want to achieve losing weight to stay healthy or to look better. The problem is that many expect to see a progress occur in a linear fashion but what they don’t realise is that it takes time, effort and patience to achieve their goals. A common cause as to why they do not see progress is because they are not tracking how much calories they have consumed. Had a smoothie for breakfast? You might think of it as healthy eating but did you know how much calories it has? On top of that you are just going to be starving a few hours later. What is the point of working out and eating well if you do not know how much calories you have eaten for the day, you can use that data to compare the number of calories you burned.

**Explain why this is a problem:** If someone who doesn’t consume the right food or consumes too many calories unknowingly, they will never be able to lose weight. They will be stuck in an endless cycle of diet and exercise with no progress. This can take a toll on a person’s psychological wellbeing and could develop into mental illnesses like anorexia or bulimia.

**Discuss the potential impact of your prospective solution:** My program aims to help people track their weight loss progress along with able to track their calories and their daily nutritional intake. The key to losing weight is consistency and diet and my program helps the user with their diet. In fact, many studies show that recording your food intake is a very effective way to help lose weight. The National Library of Medicine found that weight loss programs incorporating calorie counting led participants to lose around 7 pounds (3.3 kg) more than those who did not. It seems that the more consistently you record your intake, the better. Tracking your calories can help you identify which eating patterns you need to be mindful to successfully lose weight.

**Research:**

MyFitnessPal



Myfitnesspal is a software already used by millions world-wide. It allows users to input what foods they are eating, and it then calculates the macros for the foods as shown in the image. This is an essential feature that I plan to put in my program. In the screenshot above it shows the basic macros for grilled chicken and I plan on doing something similar like this and implementing it into my program It is available as a website or as an IOS or android app

Graphical user interface, text, application

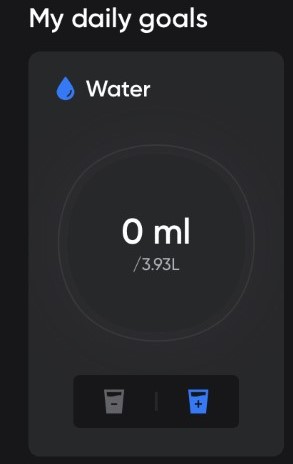
Description automatically generatedI like here how it shows all the different meal times in the food section. It also shows the user how many macros they have consumed in that day. This is another essential feature I plan on implementing as its really useful for the user to know and also its one of the main purposes of a calorie counting program.

Graphical user interface, application

Description automatically generatedIn this screenshot you can see the workouts section. Its very basic, the user is instructed to enter the workouts they did with the total amount of minutes and calories burned. While this is a great solution. I will implement this and make it better in my own way. I will add strength training and cardiovascular workouts onto my program so the user can see them and learn workouts. They can then add how many calories they burned on the main interface. Dsadsads

Nutri Coach

Available as a mobile app only. Similar to MyFitnessPal it also allows users to input what foods they eat, and it adds it to their meal log. I personally use this app and this app was the reason I decided to create a Calorie Counter App for my project. Looking at the image, it also has a drink water goal and I plan to implement this into my program. The reason I like this example is because it is beginner friendly as the UI is simple. Regarding the simplicity of the UI, I plan to make my program similar. I like how for the eaten and burned there are icons along with the number of calories and for the macros there is 0/191 etc. This is how I intend to make my program. I want to create the same icons for my program. I personally do not like the dials because it feels too generic so I plan on just putting the macros without any dials on the main interface.

 Here is a more in-depth look at the water section. I like the water glass icons with the plus and the minus, and I will be implementing this into my program. Its simple and easy for the user.

**Stakeholders:**

A stakeholder of the program will be the user who wants to lose weight. The stakeholders will enter how many calories they consumed for the day along with any supplements they have taken and this will all be logged for them. The interface will show them how much carbs, protein and fat they have consumed for the day, how much calories they burned and how much calories are left for them to eat. Each user will have their own user account so they can track their own progress. The project will be appropriate for the users need as it will be able to run on any Windows computer that the user might already be using and can be integrated into their work if they are on break. They can quickly track the calories they have consumed on their work computer.

Another stakeholder of the program will be the user who wants to bulk up and become muscular. They can use the workouts section of MyCal. The ordinary user also has the workouts section.

**Interview Analysis**

Before development of the project, I interviewed two main stakeholders:

* A user who wants to lose weight and to track his calories. (Saul Bright)
* A user who wants to track how much they have consumed for bulking (Jackie Welles)

I wanted to see what they would like to see in the program along with any features that they would consider essential. I interviewed these two people as they will most likely be using different parts of the project.

I asked these questions to Saul Bright and Jackie Welles:

1. Have you used a calorie counter app before? If so, what features did you like about it?
2. Would you place more importance on how aesthetically pleasing it looks, or on how easy it is to use?
3. To what extent do you agree with the statement “Counting Calories are like a chore”?
4. Do you want a feature where the program tracks your water intake?
5. Say if a specific food item wasn’t available, how would you fix this?

Saul’s Answers:

1. I have not used a calorie counter app before. But if I was going to use this one for the first time, I would want it to be a useful application to help me count my calories.
2. Both matter. People are going to be using an application where it provides functionality, while remaining easy to use for the user and making sure it looks good. No one wants to use a grey app with no colour and no modern font.
3. To some extent I agree with that statement. That’s why I never considered on downloading such app anyway.
4. It seems like a nice feature to have, that way I am able to see how much water I drank for the day.
5. I would want the developer to implement a feature where the user can add custom foods or nutrients to the daily log.

Jackie’s Answers:

1. I have, one feature I liked is how the app changes the daily caloric requirement to fit the needs of every user. For example, if a person wanted to maintain their weight and did a lot of activity the app would change its algorithm to meet the needs of the user. Same with if a person wanted to lose weight and did little to no activity a week
2. I would say they both matter. Aesthetics and the looks of the program will catch the user’s eye when trying to find the right calorie counting app but it has to be simple for beginners as well as give existing calorie counter users a sense of familiarity when using MyCal.
3. Absolutely not. I believe that if you count what you eat you are more likely to discipline yourself and be conscious about your macros.
4. Yes, with calorie counting there usually should be water tracking too. It is a nifty feature to have because with it I am able to be getting my recommended daily water intake. After all the human body is made up of 70% water.
5. I'd like the developer to include a function that allows users to add special foods or nutrients to the daily log.
6. An extra feature I would want in MyCal is if there was a workouts section. This will be a great feature because people who are counting their calories whether that be they are losing weight or building muscle can view a variety of cardio and strength training workouts. This feature is useful and special because, as far as I'm aware, not many calorie counting apps offer it.

**Interview Reflection**

In light of the feedback from the interviews, I want to include many of the features that the stakeholders requested. Jackie's specific request about the exercises will also be implemented. It does sound like a good idea to include workouts in a calorie tracking software because it fits well with it. I intend to add all necessary features that Saul and Jackie both asked. I want to build a user interface that contains everything needed, is not daunting or perplexing for new users, is also familiar to those who are already familiar with calorie tracking, and also has a mainstream, flagship design that most users have become accustomed to.

**Essential features**

One essential feature that I plan to put into my program is the login area. Allowing users to log in to their account. When the user enters their login credentials and clicks the login button, the program will communicate with the user database and check to make sure if the details are correct. If so, the user is authenticated, and the main window opens. Otherwise, the user is told that the login credentials specified are incorrect and to try again. This is essential as user login is used to authenticate the user, give permissions, and load data that is specific to the user.

Another essential feature that I plan to put into my program is the food database. This contains all food items that will be part of the meal log. When the user searches for a food item in the search box, the food item along with its nutritional information will all be on the database that way the user can find it and add it to their meal log. The database is very important.

One more essential feature that is needed in my program is my algorithm which determines how much calories a person should consume. There are lots of factors to this, these being height, weight, gender, if they are using the app to lose weight or bulk up weight, how much activity they do during the week. Say if a person wanted to update their stats, when they do, the algorithm will have to also amend to their needs and tell them the correct amount of calories to consume in a day.

Lastly would be the meal log. As shown on MyFitnessPal you add the foods you have consumed and this can be added to either breakfast, lunch, dinner or as a snack. It is then recorded as either one of those.

**Limitations**

One limitation that I may face is that I will not be able to create an optimal solution due to limited time. I won't have enough time to completely design all of the features that a user could want in a program. Since I'm not an experienced programmer, my programming ability may limit what I can accomplish. Another limitation is that there is currently no system in place to recover the user's account. This is due to the fact that privacy is our top priority. Because all of the users' data is held locally and no data is sent to us, I have no method of verifying the identity of anyone attempting to restore a user's account. I'd have to design a password reset API to allow users to recover their accounts, which is beyond my expertise and therefore would take too long to develop.

Another limitation is item quantity. I won't be able to include a wide selection of foods from every category in the app, so I'll focus on the most well-known foods and items that people would usually consume when on a diet. To help find a way to overcome this restriction, I'll add the possibility for users to add unique meals with their custom macros to add to their meal log.

Lastly, a limitation of my program is a password reset function. Users will be unable to request to reset their password. As our program is privacy based, we have decided not to implement this feature as we want the user to be in full control of their data. We are closely following GDPR regulations to adhere to this.

**Computational Thinking:**

Abstraction:

The process of abstraction involves eliminating unnecessary and pointless details. This enables the user to concentrate just on the most crucial aspects. This is something I would incorporate into my project because it's crucial to making my software as accessible to everyone as possible as well as the fact I would prefer for the screen to be clean and minimalistic.

An example of abstraction would be when the user would not have to see the process of the new login credentials being stored on to the database system, this would be a process that would run in the background as soon as the user has clicked on the button “Join MyCal”

Decomposition:

Decomposition is the process of breaking down a problem into smaller, more manageable problems which can be solved individually. Without the usage of decomposition, it would be very challenging to try to code this project. By breaking it into smaller tasks, I am able to make it feel less overwhelming for myself. For the user, the main interface will be the main hub for all the other features in my program. The settings icon will be on the main interface as well as the workouts section button and the water log. Once I make the main interface I am able to work on the other parts of my solution such as the workouts section, the settings section etc.

Thinking Procedurally:

The program follows in a linear and sequential order. From login to settings the order is as follows:

User has a choice to either login or register, Once the user logs in they are able to navigate to the settings section.

If the user has not registered an account, they need to enter their email, password and full name. The database will check if any records with those exact same fields match, if so then the user will be instructed to change details. Once done they will be greeted with first time start-up. Only then are they able to access the main interface and go on the settings section.

Thinking Ahead:

The process of identifying any pre-requisites that are needed before a process is carried out. Before allowing my users to use the food logger they will need to input their height, weight and goal weight. This is to ensure that my program gives the correct number of calories the users need to consume to see progress.

If a person wanted to change their goal, they would also have to update their stats to allow the program to adjust to their needs accordingly.

Thinking Logically:

Thinking Logically would be where to use branching and iteration to solve a problem. Ensuring using the best approaches to solve each problem to allow my program to create the best outcomes. Understanding the complexity of my algorithms and whether any additional steps are needed to increase complexity or decrease complexity to ensure that I can write the code within the time limitations that are currently in place would be another way that thinking logically would be beneficial to me.

To provide the user with the right experience, my program requires many parameters. Say, for instance, that my programme instructs individuals to eat more calories if they wanted to lose weight. It wouldn’t make sense. In order for my programme to function according to the users needs, the user will have to explicitly tell it that they wish to lose weight.

When the user logs into their account, the code will verify if that user exists on the database, if they do then access will be granted to them. If they don’t they will be instructed to create a new account to use my program.

**Hardware**

My program will run on PC and will require minimum specifications of today’s standard.

* Monitor (minimum resolution of 1360 x 768) – A monitor is required to display the program’s main interface and other forms to the user.
* Mouse – Able to navigate the program, and answer questions needed to determine their daily intake. The mouse can also be used to click on buttons, dropdown menus etc.
* Keyboard – The keyboard can be used to enter information such as adding to the meal log, configuring weight and height.
* 2.4GHz Processor – A CPU of this speed is required to run MyCal without any issues. The program might be able to run on CPUs slower than this but might have impaired performance.
* 5GB of disk space – This is enough capacity to store the application on the device. This should be sufficient for storing all of the data required by the software and database. This gives enough room for the database to grow, but there should still be enough storage space.
* 4GB of RAM – This is sufficient to run the software. My programme will incorporate certain memory-intensive features, such as the GUI, Interacting with the database thus 4GB of RAM would be ideal.

**Software**

* Microsoft Access – is a database software that stores data and information for users. A database is superior to conventional storage methods (such as an Excel spreadsheet) because it allows you to query the data and create relational tables. My program requires relational tables
* Windows Operating System – The programme will run on the Windows Operating System operating system. It also allows the software to communicate with the hardware (mouse, keyboard, etc.) so that the user may make the appropriate inputs. It must be Windows to allow it to work with Microsoft Access.
* Microsoft Visual Studio – The program will be created here and executed here. Unless I find a way to turn my program into a standalone exe file with my own installation wizard then it will change but for now it will be run on VS.

**Success Criteria**

|  |  |
| --- | --- |
| **Criteria** | **Justification** |
| Login form | The login form works correctly. Users are presented with an email and password box along with a login box. If they do not have an account they can click on a label that will take them to the register form. This is an essential feature as mentioned before. |
| Register Form | The register form works correctly. If users do not have an account they are able to create an account to use the program. Register and Login are high priority as without them the user cant gain access to the program |
| Watermark Boxes | Show “Email” and “Password” greyed out in the boxes before the user enters them |
| Progress Stores Properly | Progress stores locally on clients device. If data is lost then progress is lost. We take privacy very seriously. |
| Database table with the user details | This is a data table that will store the login details of the user. These are very important as every account will have other attributes attached to them such as height, weight, gender, full name. Each account is different with different activity levels and goals. |
| Database table with the details of different food and drink | This is a data table that will store different food and drink items. The user can use the search to add the different items they consumed to the meal log |
| Enter Height option | To allow my algorithm to determine how many calories they need to consume to lose weight |
| Enter Weight option | To allow my algorithm to determine how many calories they need to consume to lose weight |
| Drink Water Goal | Users can input how much water they have drank for the day. |
| Add Custom Macros | If a specific food is not available, they can read the label and add the macros onto their log to make sure they added that food. |
| Workout Section | Section that shows cardio and strength training workouts. This |
| Simple Interactive Menu | Menu to help them navigate the menu without any help |
| Exit button | Essential feature that allows users to exit the program |
| Back Button | Essential feature that allows users to go back and forth to other forms in the program. Example would be from main interface to settings and back. |
| Help section in settings | This is a feature I could add provided I had any spare time left. It will be useful for new users. |

**Design**

My software is extremely complicated, and I intend to split it down into smaller, more manageable chunks. The most significant elements, such as the Log, necessitate a large amount of code, and I intend to utilize subroutines to make programming easier. The various sorts of foods are objects, and their development will necessitate the use of Object-Oriented Programming. My application has sophisticated algorithms, one of which is how height and weight influence how many calories a person should consume on a daily basis.

Because I recently discovered that there are public databases that I may utilise, I may have to add API keys to my app. Because these public databases contain all of the meals that people commonly consume, obtaining an API key from them is essential for the construction of my app. I am still debating on whether I should include this as I need to ensure if I have enough time, if I don’t manage to use an API then I will make the food database myself.

Some critical components, such as the log, the login system, and others, will be broken down into modules because this is the decomposition stage. I'll work on each model separately and then put everything together to construct my first prototype. I'll begin with the most significant modules that require the most attention, and then go on to the less important modules, such as the "drink water log" towards the end. Before a final version of my program is ready, it will go through several prototypes. The first prototype will include a rudimentary working login screen with a basic but unusable user interface (meaning there will be menus and buttons, but they do not lead to anything). If my final prototype is stable and successful in the eyes of my peers and teacher (beta-testers). Then the final prototype will be released as version 1 of my program.  Rapid application development (RAD) and Extreme Programming (XP) methodologies will be used to design my program. The three fundamentals for making my software a successful working solution once it launches are user feedback, frequent releases, and reliable prototypes.

Chart, box and whisker chart

Description automatically generatedSystem Diagram:

Diagram

Description automatically generated

Program Functions:

Diagram

Description automatically generatedLogin:

This is the login screen. The user enters their username (the persons email is their username) and password. When they have entered their credentials, the program will verify if such credentials exist on the database and if it does, it will grant the user access to their account and the app. If they enter incorrect credentials, there will be an error message that will say “incorrect login details please try again”. It will continue to say this until they either register a new account or they figure out their account.

Diagram

Description automatically generated

Diagram

Description automatically generatedFirst Time Startup:

This flowchart shows what happens when someone creates their account for the first time. The program will prompt the user to enter their height, weight, and the goal weight they want to reach.  My algorithm will then calculate how many calories they should consume based on whether they want to lose the weight or bulk up. This is the point at which my algorithm will begin. Height, weight, gender, and physical activity during the week are all parameters that my algorithm requires to work. My app will be incorrect if the inputs are incorrect, which is why this stage is so crucial because it defines everything else about the app.

Diagram

Description automatically generated

Search Box:

Diagram

Description automatically generatedThis flowchart demonstrates how the search box functions when a user is looking for a food item to add to their log. After they've entered the food item into the search box. The database will be searched to see if that particular food item exists. If it exists, it will display the food item as well as its nutritional information, allowing the user to add it to their food log. The search box will appear when the user decides to add food they consumed to the daily log.

Chart, diagram, box and whisker chart

Description automatically generated

Chart, diagram

Description automatically generatedFood Log:

This is the flowchart for the food log, the main purpose of my program. The flowchart starts with the user already searching the food item and having the nutritional information of the food type opened already. The item info will be very complex. It will contain calories, carbs, protein, fat along with sodium, fiber, calcium, iron, and vitamins. More nutritional information will be added once the log and the food database are up and running. Next my program will allow how many grams or ml of the food/drink they have consumed to add to their daily log. The app will then show how many macros they have consumed for the day before they go over their daily limit. Finally, a success message appears to the user to show them they have added a new food item to the log.

A picture containing icon

Description automatically generated

Text

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generatedWorkouts Section:

This is the workouts section. When the user presses the workout, sections button the program will display the workouts menu. Here they have a choice of either picking cardio workouts to do or strength workouts. In the cardio section there will be a variety of workouts such as running, jumping jacks, burpees, mountain climber, plank and more. In the strength training section, they can specify which area of the body they want to focus on. Say if they want to work on their legs, a bunch of leg workouts will show. Keep in mind for strength training the workouts will be based on weights and the users own body weight too and the reason I did this is because some people may not have access to weights or a gym. That way it is newbie friendly along and gym rat friendly.

Diagram

Description automatically generatedSettings Page:

The settings page will be used if a user wants to update their goal. Wants to update their weight or to log out of MyCal. Users can also request to delete their account here too.

Forms:

**Please note these forms are concepts and may not represent the final look of MyCal**

txtTitle this is the name of my program. It serves no other purpose.

TxtDesc: this is a simple description instructing the user on what to do.

Graphical user interface, application

Description automatically generatedLogin Screen

BtnRegister: This is the register button. If users do not have an account with us, then they are able to create a new account. Keep in mind we do not have an account retrieval service and the main reason for that is privacy.

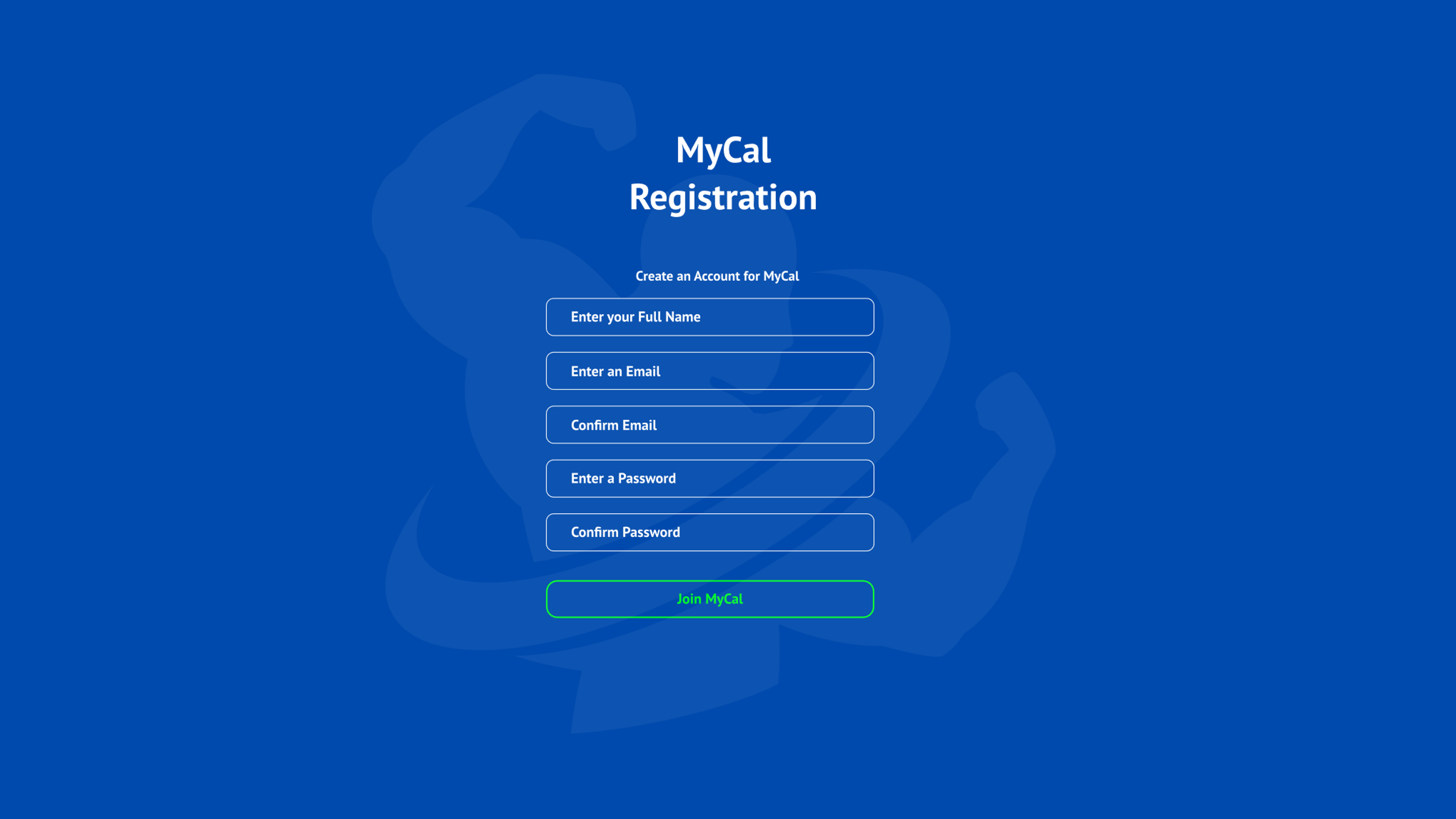
TxtPassword: This is a box that users will enter their email in.

BtnLogin: This button is used to log in to the program. Essentially It grants the user access when they enter their correct details

TxtEmail: This is a box that users will enter their email in.

|  |  |  |
| --- | --- | --- |
| **Usability Feature** | **Description** | **Justification** |
| Button | This is a button | When the user clicks on the button it will give an output. For example, when they click on “Login” it will grant the user access to the program. |
| Label | This displays text. | The “txtTitle” is a label. Its use is to output text. |
| Input Box | Box that user will enter information in. | The email and password boxes are where the user will enter their information in. These two boxes require an input. |

Register Screen



TxtTitle: Displays the title, and that’s it.

TxtDesc: this is a simple description instructing the user on what to do.

TxtFullName: To create an account, users need to input their full name here

TxtNEmail: This is a box that users will enter their email in.

TxtConfEmail: This is a box that users will enter their email in again to confirm

TxtNPassword: This is a box that users will enter their email in.

TxtConfPassword: Users will have to re-enter their password

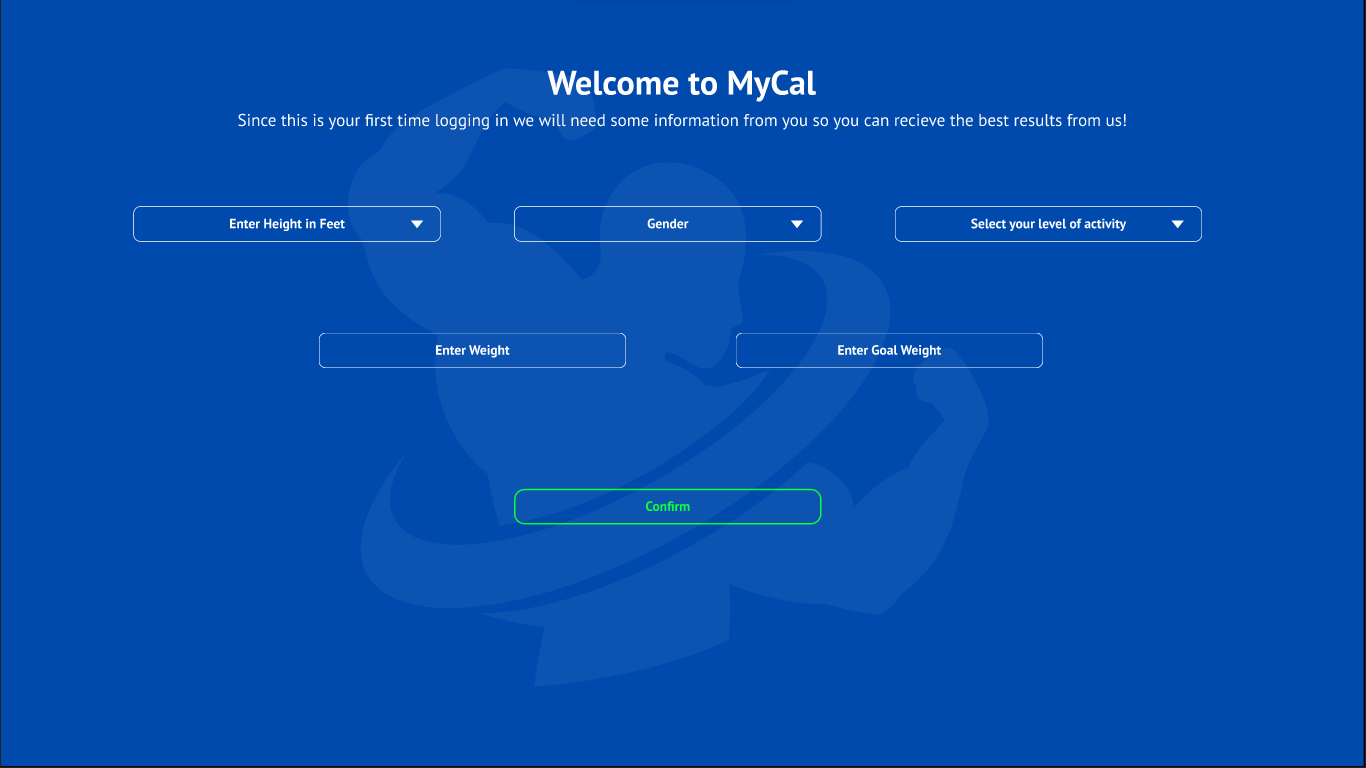
BtnJoin: When the user clicks on this button, provided they have signed up with valid information, the program will create an account for them and will store their account details in a text file in their appdata folder

|  |  |  |
| --- | --- | --- |
| **Usability Feature** | **Description** | **Justification** |
| Button | This is a button | When the user clicks on the button it will give an output. For example, when they click on “Join MyCal” it will grant the user access to the program. |
| Label | This displays text. | The “txtTitle” is a label. Its use is to output text. |
| Input Box | Box that user will enter information in. | The email and password boxes are where the user will enter their information in. These two boxes require an input. |

First Time Startup

TxtDesc: this is a simple description instructing the user on what to do.

TxtTitle: Displays the title, and that’s it.



ComboBoxActivity: A drop down menu that shows the user different activity levels for them to choose. This ranges from very active all the way to sedentary

BtnConfirm: When the user clicks on this button, provided they have signed up with valid information, the program will store their details in their new account

TxtGWeight: Users will enter their goal weight here. This is the weight they want to reach.

ComboBoxGender: This is a drop down menu that shows 2 genders to the user to choose from.

TxtWeight: Users will enter their current weight here

ComboBoxHeight: Drop down menu that shows different heights in ft for the user to choose from

|  |  |  |
| --- | --- | --- |
| **Usability Feature** | **Description** | **Justification** |
| ComboBox (Drop Down Menu) | A button when clicked displays different options for the user to choose | In my solution the drop-down menus are the boxes with the little arrows on them. For Height, Gender and Activity level users will be given choices to pick that fit their requirements. |
| Label | This displays text. | The “txtTitle” is a label. Its use is to output text. |
| Input Box | Box that user will enter information in. | In my solution, weight and goal weight are input boxes the user will use to enter information |
| Button | This is a button | When the user clicks on the button it will give an output. For example, when they click on “Confirm” it will grant the user access to the interface. |

Main Interface:

BtnSettings: When this button is pressed. It takes the user to the settings section.

PictureBoxTick: This tick indicates that the user has added their breakfast to the meal log.

TxtDesc: this is a simple description instructing the user on what to do.

Graphical user interface

Description automatically generated

BtnPlus: When this button is pressed, users can add the meal they consumed to the meal log.

TxtMacros: These display the macros they have consumed for the day. They can also see more macros when any of the macros on screen is clicked.

PictureBoxFN: This displays a vector image of a knife and fork. This is to indicate how many calories the user has consumed so far. The program tracks this.

PictureBoxFire: This displays a vector of a fire. This is to indicate how many calories the user has burned. The user manually inputs how many calories they’ve burned.

BtnWorkouts: When users click this button, it will take them to the workouts section. This section displays either cardio or strength training workouts the user can choose to do.

BtnAdd: Users can add custom macros to add to the meal log as well as set meals and burned calories.

TxtCal: Shows users how many calories they have left to consume for the day.

TxtDate: Displays the date, and that’s it.

TxtTitle: Displays the title, and that’s it.

|  |  |  |
| --- | --- | --- |
| **Usability Feature** | **Description** | **Justification** |
| Label | This displays text. | The “txtTitle” is a label. Its use is to output text. |
| Button | This is a button | When the user clicks on the button it will give an output. For example, when they click on “BtnAdd” it will grant the user to add a meal to the log. |
| PictureBox | This displays an image | In my example, there are many picture boxes. An Example is “PictureBoxFire” |

Water Log Section (part of main interface)

Graphical user interface

Description automatically generated

BtnWP: When this button is pressed. 100ML of water is added to the water log.

BtnWN: When this button is pressed, 50ml of water is taken off the water log. The reason it is 50ml and not 100 is because some people may choose to drink 1.5L or 2.5L of water.

TxtWL: This text shows users how much water they have drank on this day. The maximum amount of water they can drink in a day is 4 litres.

PictureBoxDroplet: This picture is a droplet of water. Its part of the water log and represents the water log section

|  |  |  |
| --- | --- | --- |
| **Usability Feature** | **Description** | **Justification** |
| Label | This displays text. | The “txtWL” is a label. Its use is to output text. |
| Button | This is a button | When the user clicks on the button it will give an output. For example, when they click on “BtnWP”, 100ml of water will be added to the water log. In my solution you can see the person pressed this button 5 times indicating 500ml of water was added to the water log. |
| PictureBox | This displays an image | In my example, there are many picture boxes. An Example is “PictureBoxDroplet” |

Workouts Page

TxtTitle: Displays the title, and that’s it.

Graphical user interface, application

Description automatically generated

BtnStr: A button that when clicked, takes the user to the strength workouts section. There they can perform strength training workouts

BtnBack: A button that when clicked, takes the user back to the main interface.

BtnCardio: When this button is clicked, it will take the user to the cardio workouts section where they can perform cardio workouts.

TxtDesc: this is a simple description instructing the user on what to do.

|  |  |  |
| --- | --- | --- |
| **Usability Feature** | **Description** | **Justification** |
| Label | This displays text. | The “txtTitle” is a label. Its use is to output text. |
| Button | This is a button | When the user clicks on the button it will give an output. For example, when they click on “BtnBack” it will take the user back to the main interface where the food and water logs are. |

When BtnStr is pressed:

Diagram, text

Description automatically generated

BtnBackW: A button that when clicked, takes the user back to the workouts interface.

BtnAbs: When clicked, will take the user to the back workouts section.

BtnBodyBack: When clicked, will take the user to the back workouts section.

BtnLegs: When clicked, will take the user to the leg workouts section.

BtnChest: When clicked, will take the user to the chest workouts section.

BtnArms: When clicked, will take the user to the arm workouts section.

TxtTitle: Displays the title, and that’s it.

TxtDesc: this is a simple description instructing the user on what to do.

|  |  |  |
| --- | --- | --- |
| **Usability Feature** | **Description** | **Justification** |
| Label | This displays text. | The “txtTitle” is a label. Its use is to output text. |
| Button | This is a button | When the user clicks on the button it will give an output. For example, when they click on “BtnChest” it will take the user to the Chest Workouts section where they can learn and practice chest workouts. |

Settings Page:

TxtDesc: this is a simple description instructing the user on what to do.

TxtTitle: Displays the title, and that’s it.

Graphical user interface

Description automatically generated

BtnDel: When this button is clicked it will permanently delete the users account. Upon clicking, users will receive a confirmation popup asking them if they want to delete their account. This is in place to combat accidental account deletion.

BtnBack: A button that when clicked, takes the user back to the main interface.

BtnLogOff: When this button is clicked, it logs the user off and takes them to the log in screen

BtnSave: When users have finished changing their stats, they can click this button. This overwrites their old stats and inputs their new stats into the algorithm.

TxtGWeight: Users will enter their goal weight here. This is the weight they want to reach.

TxtWeight: Users will enter their current weight here

ComboBoxActivity: A drop down menu that shows the user different activity levels for them to choose. This ranges from very active all the way to sedentary

ComboBoxHeight: Drop down menu that shows different heights in ft for the user to choose from

|  |  |  |
| --- | --- | --- |
| **Usability Feature** | **Description** | **Justification** |
| Label | This displays text. | The “txtTitle” is a label. Its use is to output text. |
| Button | This is a button | When the user clicks on the button it will give an output. For example, when they click on “BtnSave” it will save their new stats and overwrite the old ones. The algorithm will also be tweaked to adapt to their new stats. |
| ComboBox (Drop Down Menu) | A button when clicked displays different options for the user to choose | In my solution the drop-down menus are the boxes with the little arrows on them. For Height, Gender and Activity level users will be given choices to pick that fit their requirements. |

Entity Relationship Diagrams:

Diagram

Description automatically generated

Data Dictionaries:

Register:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Purpose | Validation |
| Email | Short Text | This stores the user’s email so they can access their account and see their stats. | This is to check if the entered email is present on the database. |
| Password | Short Text | The password is a security feature that locks a person’s account so unauthorised users cannot access their account. | Password must contain a capital letter and number |
| Height | ComboBox | This stores the user’s height in the stat database. | The minimum height will be 4 ft 11. The max height will be 7 ft. |
| Weight | Short Text | This stores the users weight in the stat database. | There will be a minimum weight for each height |
| Goal Weight | Short Text | This stores the goal weight of the user. This is the weight that the user wants to achieve | Goal Weight can’t be under a certain amount, for example the ideal weight for a 6 foot person is 65-77KG. A person’s goal weight cannot go under that or else they will be underweight |
| Activity Level | ComboBox | Stores the level of activity the user does in a week. Goes from Sedentary to Most Active. | Will only contain 5 levels. |

Login:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Purpose | Validation |
| Email | Short Text | This stores the user’s Email so they can access their account and see their stats. | This is to check if the entered email is present on the database. |
| Password | Short Text | The password is a security feature that locks a person’s account so unauthorised users cannot access their account. | There will be a password length requirement. Minimum length of 8 characters. Password must contain a capital letter and number |

Food Log:

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Purpose | Validation |
| Food Item | Short Text | The user enters what food item they want to add to the food log. | This is to check if the entered food item is present on the database. |
| Drink | Short Text | The user enters what drink they want to add to the food log. | This is to check if the entered food item is present on the database. |

Variable Table

Register:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data Type | Scope | Purpose | Validation |
| Email | String | Local to btn\_login, method of frmLogin | Email as person logs in. | Presence Check – Error is displayed when nothing is in the email box. |
| Password | String | Local to btn\_Login method of frm\_Login | Password as person logs in. | Length check – Minimum password length of 8 characters. Password must contain a capital letter and number |
| Height | ComboBox | Local to btn\_Confirm method of frm\_Register | This stores the user’s height in the stat database. | Length Check -The minimum height will be 4 ft 5. The max height will be 7 ft. |
| Weight | Short Text | Local to btn\_Confirm method of frm\_Register | This stores the users weight in the stat database | There will be a minimum weight for each height |
| Activity Level | ComboBox | Local to btn\_Confirm method of frm\_Register | Stores the level of activity the user does in a week. Goes from Sedentary to Most Active. | Will only contain 5 Levels |

Login:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data Type | Scope | Purpose | Validation |
| Email | String | Local to btn\_login, method of frmLogin | Email as person logs in. | Presence Check – Error is displayed when nothing is in the email box. |
| Password | String | Local to btn\_Login method of frm\_Login | Password as person logs in. | Length check – Minimum password length of 8 characters.  Password must contain a capital letter and number |

Data Structures:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Type | Scope | Purpose | Validation |
| Email | Access Database  (table) | This is the primary key for the User table in the database. It is unique to each user. | Email as person logs in. | There cannot be an email that has been registered twice. |
| Stats | Access Database (table) | The stats will be stored in the User table. As each user is unique, so is their stats about their body | Contains persons height, weight etc. | Stats will have to be entered; they cannot be left blank in the database. |
| Food/Drink Item | Access Database | Different Foods and Drinks will be stored in the Food Table. Stored in the access file but in a different table. | Users enter the food and drink they have consumed for the day to their meal log. | No validation needed. |

Class Diagrams:

Diagram

Description automatically generated

Test Data for Development:

Login (email)

|  |  |
| --- | --- |
| Test Data | Type |
| [John\_Doe@outlook.com](mailto:John_Doe@outlook.com) | Valid |
| [JohnDoe5798437578439075@outlook.com](mailto:JohnDoe5798437578439075@outlook.com) | Valid Extreme |
| JohnDoe\_outlook.com | Invalid |
| johndoe | Invalid Extreme |
| T | Erroneous |

Login (password)

|  |  |
| --- | --- |
| Test Data | Type |
| Strong78 | Valid (minimum 8 character password required) |
| 9UUFtKak1mP3zPe766WvBPikLQXdhgno5 | Valid Extreme |
| mycatsnameisjoe | Invalid (password must contain a capital letter and number) |
| rhhgjkdfhguify9768u095876ytijgrj | Invalid Extreme |
| # | Erroneous |

Height

|  |  |
| --- | --- |
| Test Data | Type |
| 6 ft | Valid |
| 7 ft | Valid Extreme |
| 3ft | Invalid |
| 1ft | Invalid Extreme |
| 234234ft | Erroneous |

Weight

|  |  |
| --- | --- |
| Test Data | Type |
| 60Kg | Valid |
| 300Kg | Valid Extreme |
| 20Kg | Invalid |
| 1Kg | Invalid Extreme |
| A | Erroneous |

Nutritional Information

|  |  |
| --- | --- |
| Test Data | Type |
| 500 Calories | Valid |
| 3000 Calories | Valid Extreme |
| -1 Calorie | Invalid |
| 9999999 Calories | Invalid Extreme |
| No numbers | Erroneous |

Test data for final testing

|  |  |  |  |
| --- | --- | --- | --- |
| Test number | What is being tested | Test data | Expected Result |
| 1 | Weight is being tested.  The user cannot enter a weight that is way too low. It is not possible for a human to weigh 5Kg | 5Kg | Should produce an error message telling the user that their weight is too low. |
| 2 | Email is being tested | [saeedurrahman@outlook.com](mailto:saeedurrahman@outlook.com) | If the email exists on the database then it should allow the user access once they have put in their password and press the “Log In” button |
| 3 | Height, the maximum height this program supports is 7ft | 8ft | An error message should be displayed saying “Height is not supported with MyCal” |
| 4 | Password, password must contain a capital letter and a number | easypasswordtocrack | An error message should be displayed because the requirement for the password |
| 5 | Food Item adding to the log is being tested. The food item must exist in the database or an error messasge will display | Doritos | An error message should be displayed stating that the food item does not exist in the database. |
| 6 | Drink Item adding to the log is being tested. The drink must exist in the database or an error message will display | Fanta | An error message should be displayed stating that the drink item does not exist in the database |

Add more as you go

**Development**

**Prototype 1**

­­­­Prototype 1 consists of a Loading Screen, a Login Form and that leads to the main interface which hasn’t been designed yet but exists as an empty form when logged in.

**Prototype 2**

Prototype 2 consists of the main interface and a working register screen that signs up a user with all their details

**Prototype 3**

Prototype 3 consists of a working settings screen and workouts section

**Prototype 4**

Prototype 4 consists of the working meal log system

**Prototype 5:**

Prototype 5 consists of the working water log system

Loading Screen:

My first prototype is a loading screen that appears on startup and this leads to a login screen. At the moment the loading screen serves no functionality other than aesthetics.

Icon

Description automatically generated

Here is my loading screen. It is a simple background with a blue loading bar

You can also see in the loading screen picture that there is no exit or minimize or window buttons on the top right of the form. This is because I removed this in the ‘Form Border Style’ in the properties section.

A screenshot of a computer

Description automatically generated with medium confidence

These are the properties of the loading screen. As you can see the ‘BackgroundImage’ is my custom design I made on photoshop.

When it is run it appears in the middle of the screen. I configured this in the properties section. You can see it here:

A picture containing graphical user interface

Description automatically generated

Here is the loading screen when it is executed.

Graphical user interface, application

Description automatically generated

As you can see it launches successfully. The loading bar works and there are no borders and it also appears in the centre of the screen. I have also created icons for each form and for the loading screen form you can see in the taskbar there is a MyCal logo icon instead of the default visual studio icon.

Graphical user interface, application

Description automatically generated

Here is the code for the loading screen.

The way for my loading screen to work is to add a timer.

Text

Description automatically generated I added a timer from the toolbox and here is the code for it. The actual loading bar will increment by 10 like this:



After 100 milliseconds,



Once the loading bar is full then the timer will be disabled. It will then connect to my other form which is the login form. The loading screen will then disappear, and the login form will appear. You can see this in the code screenshot above.

Login Form

This is my next form that is part of prototype 1. Once the loading screen is done it goes to the login screen. Here is a picture of my login form.

A screenshot of a computer

Description automatically generated with medium confidence

It has a simple email and password login box with the login button. There is no register button as of yet but I plan to add that in prototype 2.

I added a feature where before the user enters their email and password. It shows a greyed out version which indicates to the user where to enter their email and password.

Text

Description automatically generated

When text is entered. The watermark disappears.

However the downside to this is that for the password box, it doesn’t allow me to hide the characters:

Graphical user interface, application

Description automatically generated

It shows the password. This is a problem as this box is usually meant to be a string of asterisks to hide the actual password.

If I put in the “PasswordChar” property then this would happen:

Graphical user interface, application

Description automatically generated

You can see that the watermarked text becomes asterisks.

Im first going to start off with the **email box**. Here is the code for it:

Text

Description automatically generated

So line 34 is an if statement, it states that when clicked, if the watermark text “Email” is still there, It will be removed in line 35 to null and line 37 states that the text colour from that box will change to black. The watermarks original colour is Silver.

A screenshot of a computer

Description automatically generated with medium confidence

Here is when the user clicks off the Email box. If there is nothing entered in the email box (line 44) then the if statement will change the empty box back to the watermark “Email” and this can be seen in line 46. In line 48 it changes the colour from black back to silver for the watermark.

Text

Description automatically generated

The same is applied for the Password box.

Here are screenshots of my Access database that contains my logins.

This is the design view:

Table

Description automatically generated

Email and password are both short text.

Here it is in datasheet view:

Table

Description automatically generated

“admin” and “s212658” users work however “test” still does not work. I am actively trying to fix this issue as this will interfere with registration and new accounts wont be able to login then.

Text

Description automatically generatedI connected my access database to my solution in the data sources here:

I am using a Microsoft Access database. It contains one table and that is “tbl\_users”

Graphical user interface, text, application, email

Description automatically generated

This is the .xsd file in the access database. It allows me to input SQL statements to allow the database to connect with my solution.

Graphical user interface, text, application

Description automatically generated

This is the SQL statement for “GetDatabyEmail…”.

This is the actual SQL statement that connects with the login form. Its selecting the email and password fields from the users table.

WHERE [Email] = ? AND [Password]=?

The question mark is a variable binding. It will prompt for email and password after submitting the query. It's telling the database "I am going to give you data, put that data in this spot". When the data entered by the user matches with the data that is present in the records of the table then it will grant the user access. A drawback to this in my solution is that the user is able to input unsanitised data. Unsanitised data regarding to my solution for example is when a password that has spaces or emails that do not have an @ symbol in it go through.

Text

Description automatically generated

Here is the code that allows access to the main interface when the persons login details are correct. Lines 79 to 84 mean that if a person has not entered anything in the email box then it will show a message box saying that their login details are incorrect. They will then be prompted to try again. Lines 87 to 88 are what connect the database and its user table to the login form. It gets the Email and Password fields and the user enters the data in the email box and the password box. This connects to the SQL statement mentioned above: “WHERE [Email] = ? AND [Password]=?”

If successful then it will allow the user access into the main interface. The Login form will be hidden and the “MainInterface” form will show. If they get it wrong then Line 101 comes into action.

Lets try both the admin and the s212 account to see if they work

As you can see the admin login works:

Graphical user interface, application

Description automatically generated

And for the S212658 Login:

Graphical user interface

Description automatically generated

But the “test” login does not work:

Graphical user interface

Description automatically generated

As you can see all message boxes work successfully.

When the user clicks on the OK button then it leads them to the main interface:

Shape

Description automatically generatedThis will be created in prototype 2 but will have no functionality on it.

I have also added an X button on the top of the login form

A picture containing text

Description automatically generated

Here is the code for it:

Text

Description automatically generated

When the X button is clicked it stops the program. That’s all.

Prototype 2 Plan:

Create Registration form

Create main interface but it serves no functionality.

Fix the look of the login button and the text boxes (make it more aesthetically pleasing).

So first I didn’t like the initial design of the Login button so I decided to create my own button class. I watched a few youtube videos to help me start. This took around 4-5 hours to complete due to my understanding of C# being very base level.

It took me 172 lines of code to create this button:

Graphical user interface, application, website

Description automatically generated

It looks almost identical to the original concept and I’m very happy it works. If you want to see the code for it I will put it in a .txt file called “class for button.txt” in the documentation folder as its way too long for documentation here. When the mouse hovers over the button it highlights like this: Graphical user interface, application

Description automatically generated

However an Issue I came found with the button is that there is a small white outline on it. I still don’t know how to resolve this. By looking at the border radius and editing the Y value of the white inner box but it still didn’t solve it.

 This was the old button.

Here is the main interface:

Graphical user interface

Description automatically generated

The form is working properly but still needs functionality. I am really happy how this turned out. By using my custom button class I was able to make the main interface identical to the concept. I still need to add the settings button on here.

The Registration Section:

Here is the first version of my registration screen:

Graphical user interface, application

Description automatically generated

It is very simple but it serves no functionality.

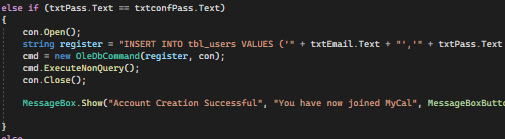
Here is the code:

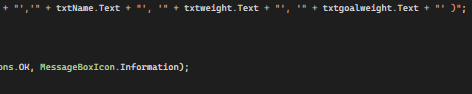
A screenshot of a computer

Description automatically generated with medium confidence

This line states that if the text boxes for name and email and pass are empty or there are no characters entered then it would show a popup saying that the fields are empty. The reason I did this with these variables only and left the rest is because these variables are the most important in the database. They grant access to each user. All the other variables (excluding confirm password) can be changed later in settings.

Going back to Confirm password:





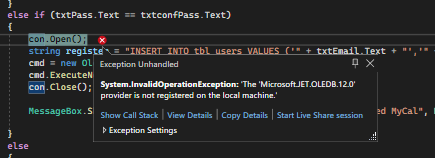
Images were split so you can see the code better.

As this is an if statement. In the else, it says that if the confirm password and password are equal then it will start an SQL statement.

The SQL statement states to insert all the concatenated variables into the user table in the database.

When its done it displays a message box stating that the account was created successfully and the user must then go back to the login page to continue.

When I ran my program and entered the registered details I had this error.



I didn’t understand what this error meant so I decided to do some research and read on the documentation for C# when trying to implement an access database into a program.

I first found that the error originated from this line: 

I then found that JET couldn’t provide a 64 bit driver for my 64 bit machine. After doing more research I found that the new Microsoft ‘ACE’ database engine does provide the 64 bit driver and includes backwards compatibility for all .mdb and .accdb database files. This was perfect for me. However I am very annoyed that I spent a lot of time trying to fix this issue.



I now changed it to ace.

Graphical user interface

Description automatically generatedAs you can see it works successfully.

I still need to add the combo boxes Height, Gender and Activity Level.

I decided to change my design a bit

Timeline

Description automatically generated with medium confidence

It looks a lot more aesthetically pleasing and I also added the combo boxes for Height, Gender and Activity Level

For height the values are like this: 6 ft 1”.

For the activity level I know that users will be confused about what this meant so I decided to add a little ‘information’ button. The numbers for the activity level go from 1 to 5, here is what it looks like when the user clicks on the button.

Graphical user interface, application

Description automatically generated

Initially it was the same blue background as the register screen however it looked very confusing as there was no outline on both the register form and the information button form. So to fix this I made the blue colour of the form for the information button lighter. It looks much better and the form also stands out. To make it easier for the user I added a little X button on my form and it closes when it is clicked. I added some extra detail to all forms so when a user hovers their mouse over a button it becomes the hand icon or when a user hovers their mouse over a text box it becomes an I button.

As now there were combo boxes added I tried adding them to my concatenated SQL string however I always ended up with this error:

A screenshot of a computer

Description automatically generated

There was a syntax error in the sql statement. I didn’t know how to fix this so I decided to look at the documentation again and read some stack over flow solutions. I found that creating parameters instead was way better then concatenating all my variables in a single set of parenthesis.

So I did just that.

Text

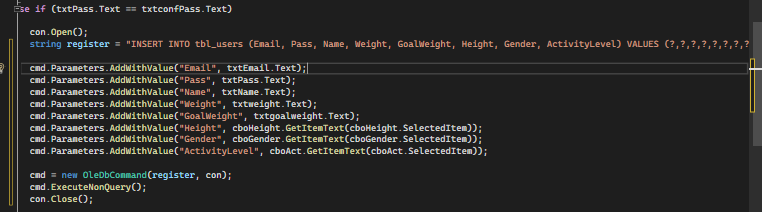
Description automatically generated

I created parameters for all the values and when I ran the program I still got an error:

Text

Description automatically generatedNow I had no idea what this meant. I didn’t understand why it was a missing operator when it was clearly defined in the string.

After doing more research I found that OLEDB does not support named parameters in a SQL string and that I have to remove every character and only leave out question marks like this:



This time I included the key values in a separate parenthesis and then the actual variable values as question marks to make sure all data fits in the correct position in the database.

Even after doing this I still got this error:

“System.Data.OleDb.OleDbException: 'No value given for one or more required parameters.”

Now I was very confused and decided to ask for help online. I got feedback and I found that I am instantiating after the parameters and losing them.

I was instructed to add the parameters like this:

cmd = new OleDbCommand(register, con);

//then the values:

cmd.Parameters.AddWithValue(“key”,value);

...

//then execute...

cmd.ExecuteNonQuery();

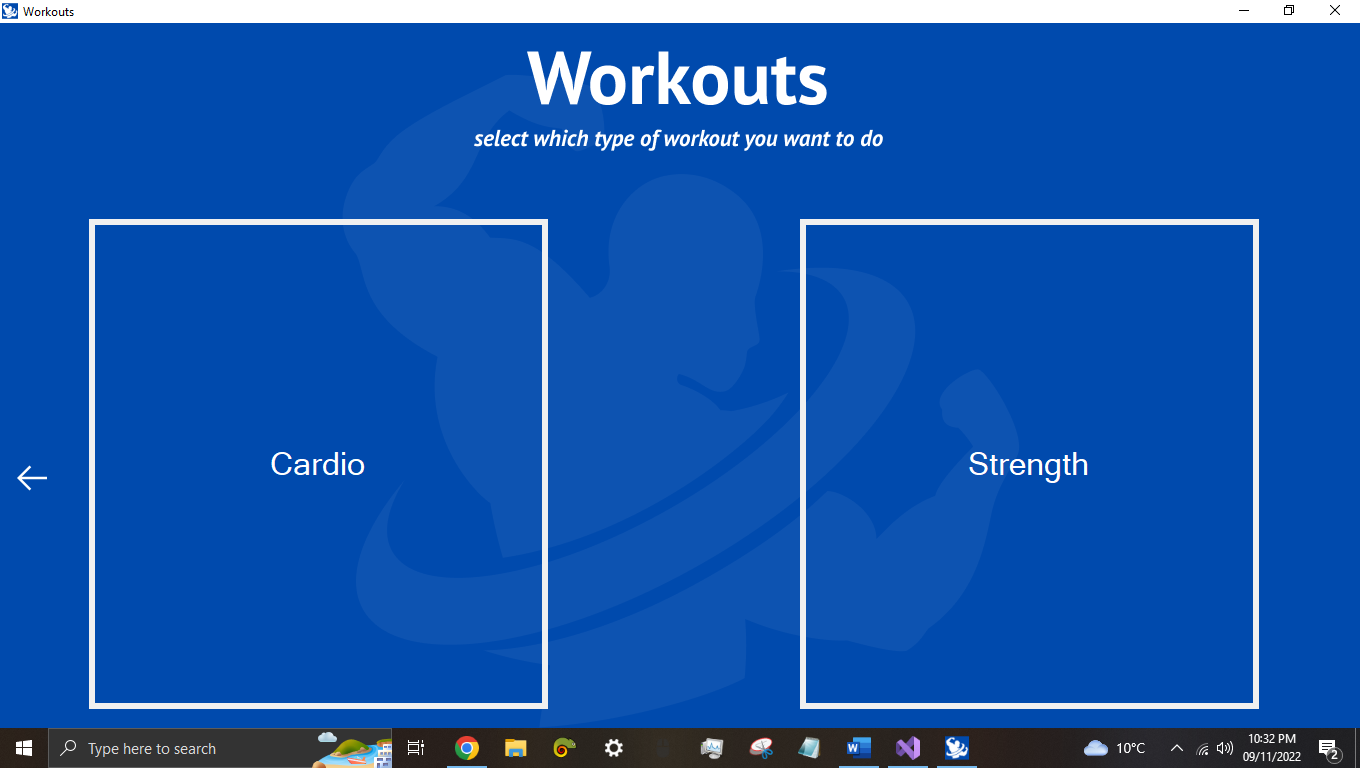
Code goes here

Prototype 3:

-Working Settings Screen

-Working Workouts Section

Here is the Workouts section:



When the user clicks on the workouts button on the main interface they are prompted to choose either between cardio and strength workouts. This form looks almost identical to the concept. However I ran into many encoding issues with the vector files for the strongman background and the top text. As figma vector files just didn’t work with Visual Studio I switched to lossless png files.

A picture containing calendar

Description automatically generated

This is the cardio section. There are gifs to help the user understand the workout as well as a back button to return to the workouts section. Its very minimal but informative. I used winform panels to help me do this. Due to limited time constraints and to save storage I was unable to do a lot of cardio workouts.

Graphical user interface

Description automatically generated

**fdfsfdsfsdfsdfsdfsd**