**MyCal**

**Analysis**

**Describe the problem:** When people want to lose weight, they usually eat 1 meal a day and practice fasting just to see a decrease on the scale. This **is** a method people can use but when they do this, they usually forget the important nutrients such as vitamins and amino acids that are just as important for weight loss. While you can get your nutrients from supplements. Studies have shown that the body absorbs nutrients more efficiently when it comes from food. Malnourishment is a very common issue when it comes to dieting and when practicing calorie deficit. Consuming 1 meal a day is enough to go for the day, but it must be very nutritious so that the body functions at its peak during exercise. Another problem that peoples face when trying to diet is they often consume foods that they perceive to be healthy. People may consume too many starchy foods that are calorie dense and may follow the ideology that being full of carbs helps with weight loss when this is completely wrong.

**Explain why this is a problem:** If someone who doesn’t consume the right nutrients or consumes too many calories, 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 as they are stuck to believe that they will never lose weight.

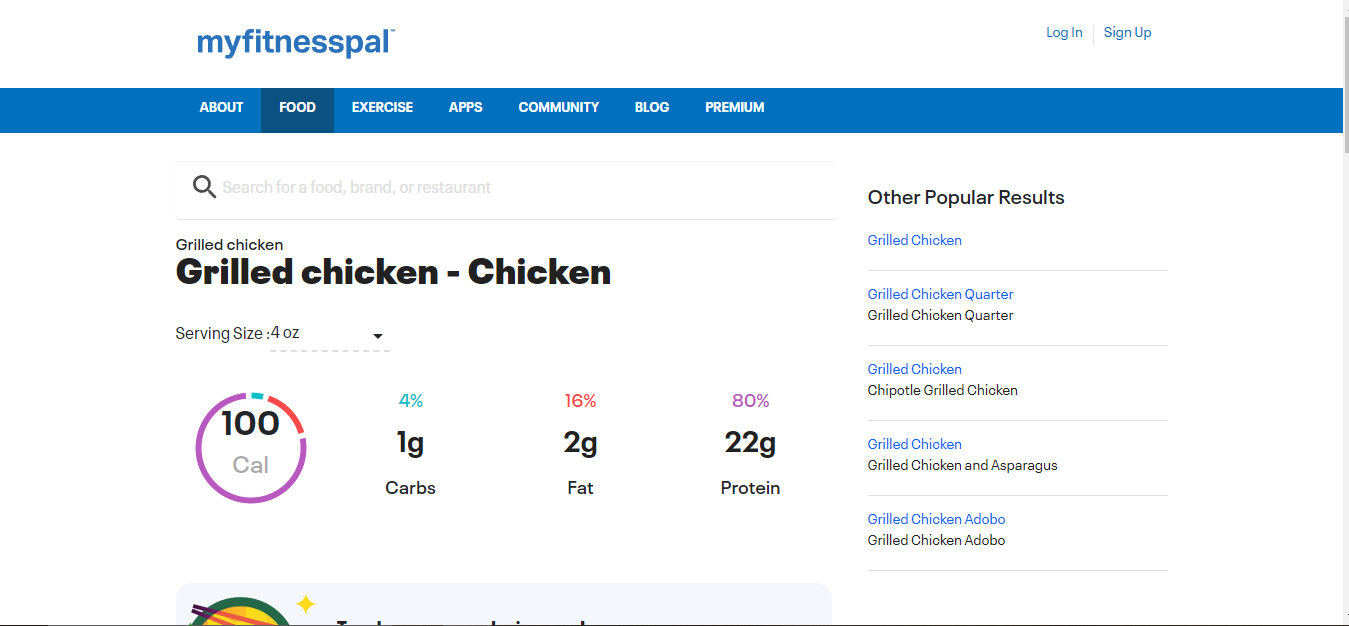
**Discuss the potential impact of your prospective solution:** My program aims to help people track their weight loss progress along with able to track their calories and their daily nutritional intake. The key to losing weight is consistency and diet and my program helps the user with their diet.

**Stakeholders:**

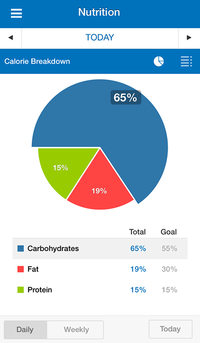
The target audience for my software is people who want to lose weight or want to track how many calories they are consuming on a daily basis. This could be anyone, of any gender, of any age but will primarily be used by people who are self-conscious of what they eat. This software will start on computers first and then slowly transfer over to mobile devices. This is because to increase portability of this program. The program should be as simple to use as possible from the user's perspective, so not to deter them from using it due to it being too complicated to use the program.

Research:

MyFitnessPal

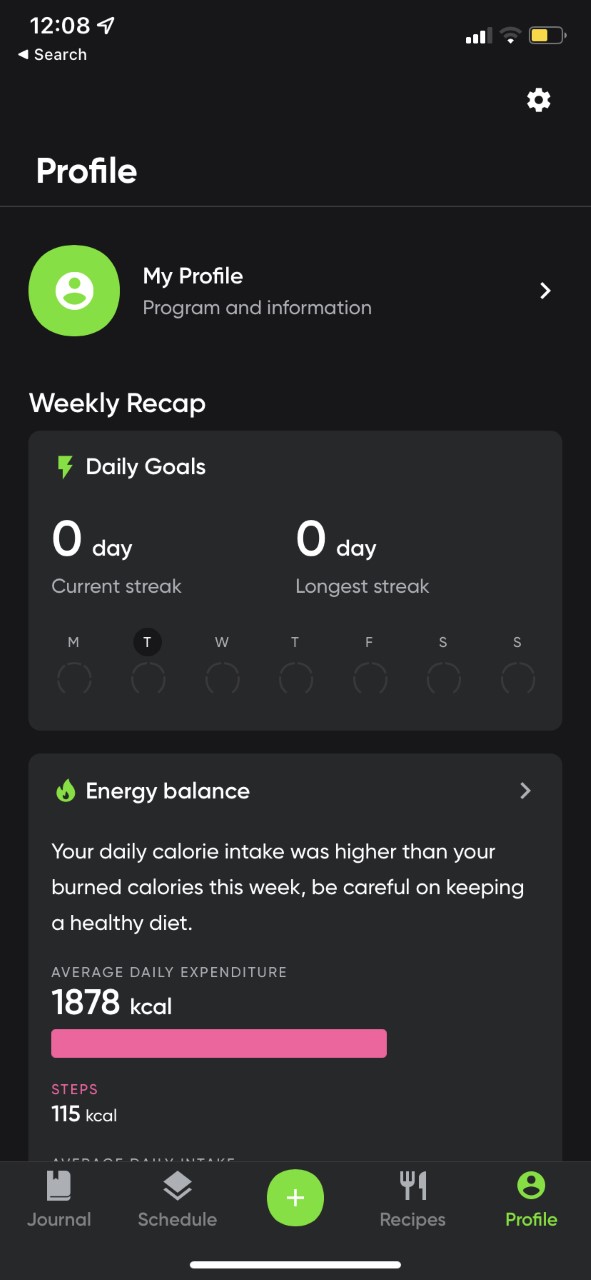


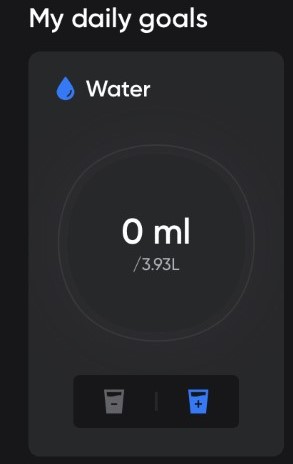
Myfitnesspal is a software already used by millions world-wide. It allows users to input what foods they are eating and it then calculates the macros for the foods as shown in the image. What I like about it is that the software keeps track of your progress toward your objectives and provides discussion forums with other users. Conversations, recipes, suggestions, and personal success stories can all be found in the forums. The reason I like this is because you are able to see how far your progress has gone The nutrition database on MyFitnessPal is huge, with over 5 million items. You can also build custom foods and dishes by downloading recipes from the internet. The software will even remember your favourite meals for easy tracking. Additionally, the barcode scanner in MyFitnessPal allows you to quickly enter nutrition information for some packaged goods. The breakdown of carbs, protein, and fat for each day is displayed as a pie chart. You can also keep track of how things went or how you felt by writing a note for each day.

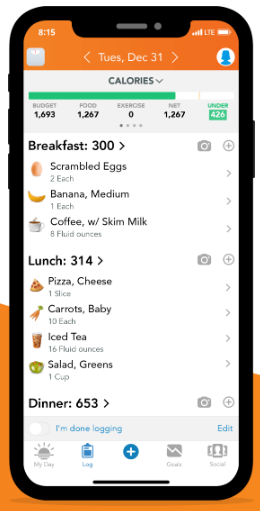
Here is a picture of the Pie Chart. We can see in this example carbs are the majority here in this users diet.

Nutri Coach

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. Instead of a pie chart, this app uses a “dial-like” UI to show you how many calories you have consumed. Personally, I think its simpler to use dial like system with exact figures of your macros.

This is the profile section of Nutri Coach. We can see that there is a weekly recap to see how many days you logged your meals and there is also a streak counter. A unique feature of this app that competitors do not have is the energy balance tab. In this example, the user ate more calories than burned and so the app lets the user know.

Here is a more in-depth look at the water section

Lose It!

Lose It! is another health tracker that includes an easy-to-use food diary and an exercise log. You can also connect a pedometer or other fitness device. Based on your weight, height, age, and goals, Lose It! provides a personalized recommendation for calorie intake. It then tracks your calories on the homepage. It features a comprehensive food database and an icon representing each food entry. The food diary is simple and user-friendly. Adding new foods is not complicated. Additionally, the Lose It! app has a barcode scanner for packaged foods, and common foods are saved for quick entry later. I like the barcode scanner feature as it can scan pre-packaged goods and to add to its macros.

**Essential features**

One essential feature that I plan to put into my program is the search box. Users can search quickly what foods they want to add onto their log. This will be implemented with a backend database that contains all the items the user wants to choose to log in their account.

Another essential feature that I plan to put into my program is the login area. Users can create accounts and their account data will be stored locally on their device so they can save their progress and so we cannot access their personal data. This prevents privacy issues as none of their data gets sent to us. When users create an account with us, we will give them a clear message on a message box on the app stating, “We do not store your data, your data will be stored locally on your device”.

One more essential feature in my program is the actual log system. The log system is what allows users to log the foods they ate, and it tracks the calories they have consumed throughout the entire day. Without this crucial feature, my program is essentially useless.

Another 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.

**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.

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.

**Hardware**

My program will first start on PC and will slowly move to Mobile devices specifically iPhone

* Monitor – Be able to view the GUI and log.
* Mouse – Able to navigate the program, and answer questions needed to determine their daily intake. Multiple choice questions will require the user to select their answer with the mouse. It will also be needed to navigate the menu and select what they want to add to their daily log.
* Keyboard – used to input answers as well as enter login information.
* 1.5GHz Processor – needed to run the software and process the data entered. This should be sufficient to execute the software quickly enough to provide a comfortable user experience with minimal loading times.
* 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, 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.
* 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.
* IOS Operating System – During the later stages of development, I will try to make a successful port of the program to the IOS operating system. There are many services which apple have themselves endorsed. These are SQL Lite, Realm, CoreData and that’s just to name Database services.

**Success Criteria**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Justification** | **Preference** |
| Users can log into their accounts | To allow them to add new items to their log | Must |
| The log works properly | The purpose of the log is to track how many calories the person has consumed for the day. If the log doesn’t work then my program is useless | **Must** |
| There is a variety of foods the user can choose to add to their log | This is so there is variation of foods to add to the log. People come from different cultures and different cultures have different foods they eat | Must |
| Enter Height option | To allow my algorithm to determine how many calories they need to consume to lose weight | Should |
| Enter Weight option | To allow my algorithm to determine how many calories they need to consume to lose weight | Should |
| Drink Water Goal | Users can input how much water they have drank for the day. They can track their water intake and see how much they drank for the week too. | Could |
| Different Sections for Different Mealtimes | It makes the UI more simpler and users are able to see how many calories they consumed for different mealtimes such as breakfast, lunch, dinner. | Could |
| Watermark Boxes | Show “Email” and “Password” greyed out in the boxes before the user enters them | Could |
| Progress Stores Properly | Progress stores locally on clients device. If data is lost then progress is lost. We take privacy very seriously. | Must |
| Additional Copies Of Progress | Users can properly copy progress files and upload them to any cloud storage service they can. That way if progress is lost they can just restore it by replacing the new progress file with their backed up one in file manager. | Would |
| An import progress system | Users can import progress if their progress got corrupt. For this to work they would have to back up their data first then during sign up there will be an option which allows users to “import progress” and there they would have to upload their progress files to the app and the app would do the rest. | Would |
| Allow users to seamlessly transfer progress from one device to another | Works the same way as the import system does but they can now import their progress to another device. All they need to do is copy the progress files. | Would |
| 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. | Could |
| Barcode System to add pre-packaged goods | Adding a barcode scanner system to allow adding pre-packaged goods to your log | Would |
| Allow users to add their own foods to the log | If a food is not available in our database, the user can add it themselves locally. | Could |
| Workout Section | Section that shows cardio and strength training workouts | Would |
| Simple Interactive Menu | Menu to help them navigate the menu without any help | Must |
| Help section in the settings | A simple guide to help them what to do if they forgot something and also show them features that they didn’t know were there before | Could |
| Interactive Tutorial | The first time they open the app, a tutorial would help them and show them how to navigate on the app and how to fully utilise all features | Could |
| Notifications | If they haven’t added their meals within a certain time frame the app will send out notifications to alert them to add their meals to the log. For example, if they had breakfast but forgot to add it, the app will send them a notification 1 hour after not adding it. It follows the mealtime clock so if they breakfast at 9am and forgot to add it, the notifications will be sent at 10am. | Must |

**Computational Thinking:**

**Thinking Abstractly:**

Abstraction is when unessential features are not shown to the user. Only the relevant features will be shown to the user. My program will implement techniques used in abstraction to make sure the inner code is hidden from the user.

Users do not need to see the actual math behind counting all the calories they have consumed for the day. They only need to see the results.

The backend database that stores users’ credentials will not be visible to the user. They do not need to see it as it may contain other users’ credentials.

**Thinking Ahead:**

The menu layout will need to be simple to use. Everything will be in its appropriate location, and users will be able to recognise how to go to the parts of the program that they want and need.

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.

**Thinking Logically:**

My program needs different arguments to give the right user the right experience. For example, say if they want to lose weight and my program tells them to eat more calories. It wouldn’t make sense. So, the user will have to state to my program that it wants to lose weight and so the program will correct itself accordingly. I will use a flowchart to help develop my system.

**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; however, building a database for mobile devices is absolutely new to me, and I plan to learn how to do so over the summer holiday.

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.

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:

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.

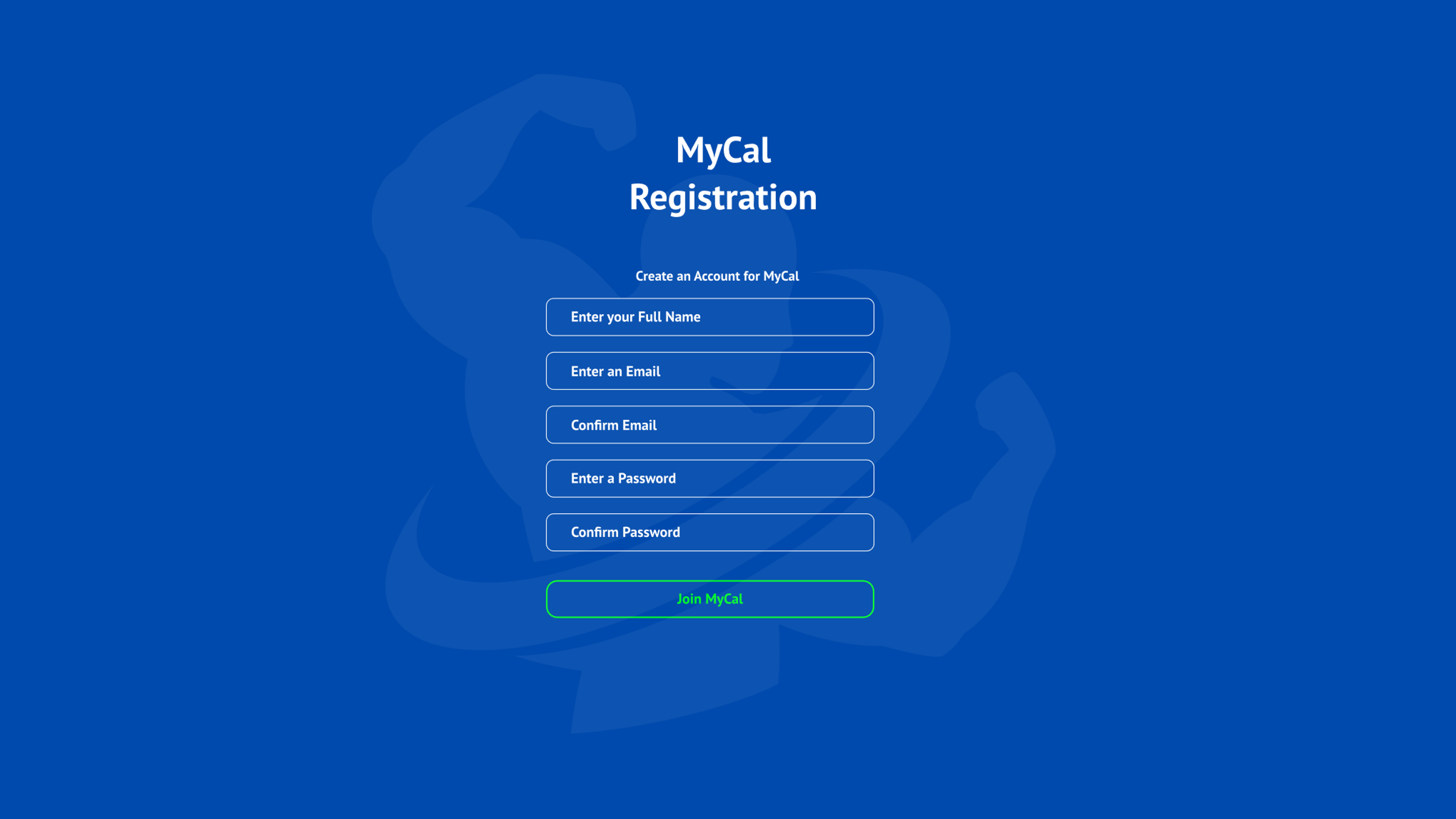
TxtPassword: This is a box that users will enter their email in.

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

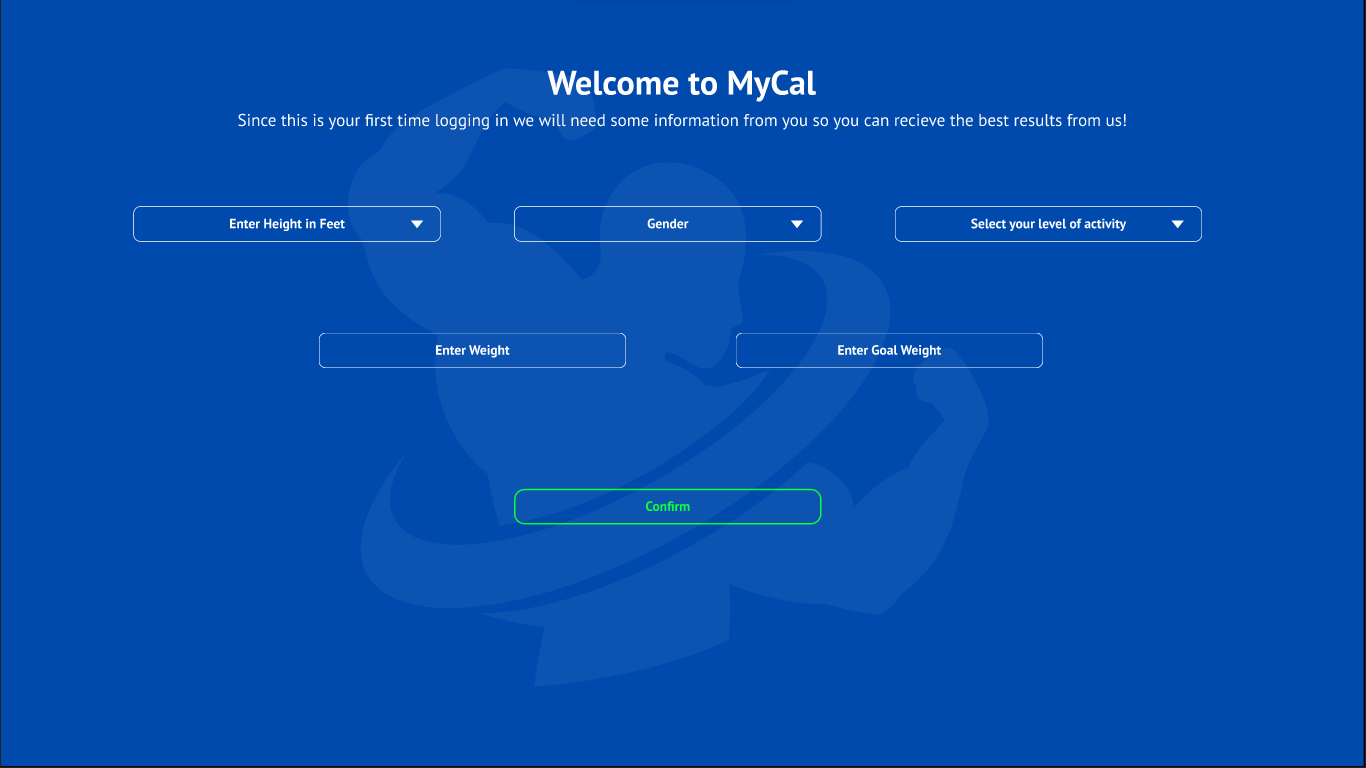
BtnJoin: When the user clicks on this button, provided they have signed up with valid information, the program will create an account for them and will store their account details in a text file in their appdata folder

|  |  |  |
| --- | --- | --- |
| **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” |

ADD FOOD SECTION HERE\*\*\*

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 5. 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.

Loading Screen:

So, 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. I am going to hopefully fix this later in development.

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.

I connected my access database to my

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

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.