**Laguna College**

**San Pablo City**

**ITE Department**

**CALORIE INTAKE MANAGER**

**January 2022**

**Instructor: Ma. Cristina Atienza Delos Reyes**

**Del Mundo, Ricsean Earl C.**

**Quiliope, Paul William V.**

**Reyes, Allen Howard M.**

**BSCS – 3**

**Abstract**

In today’s generation, being fit and healthy is a must for each one and since we are experiencing a worldwide pandemic that we should always stay inside our homes. We should always make sure that our body is fit and healthy. The goal of Calorie Intake Manager is to help in computing the calorie intake in a day for people who are cautious about their diet. It’s a great app for analyzing and managing your daily calorie intake. The hassle of searching and computing how much calorie is there in a food will be lessened because this app can do it. This app is great for people who have a strict diet that wants to follow a goal of calorie intake. It can also be used by people who just wants to check how many calories they intake in a day. This app will tell you how many calories there are in a food, and it will total how many calories you’ve intake for the whole day. You can also track your day-to-day calorie intakes. Counting calories is one way to make sure that you don’t eat too many calories or to know if you are eating enough calories. If you count your calories, it will make you closer to obtaining the goal that you want. And the best application you can use for that is the Calorie Intake Manager.

**Purpose**

The main goal of the program focuses on the health of the users. It can be used to achieve a weight goal assigned by the user. It will tell the users how many calories they need to consume for a day in order to achieve their weight goal. Furthermore, it will also tell how much calories do a specific food has, and it will also give the users a full meal suggestion for their diet. If they follow the program, they will see results, and soon will achieve their goals. This program will be a useful tool in order to maintain or achieve a healthy diet and weight for its users. It can help in preventing and also changing an abnormal body weight such as underweight, overweight, and obese. It will also help in achieving a healthier way of living, especially nowadays, where we need our bodies to be strong and have a higher resistance to sickness. Furthermore, it can motivate people to continue on pursuing their body goals because it will track and compare every change you’ve accomplished. Besides, it can boost the happy hormones of the users because seeing good results will make anyone happy. Moreover, it will help you to have a more balanced diet and healthy body because using this program will tell you a lot of what you want to know about your weight. The program also includes computing your BMI and determining what category do you fall in based on your BMI.

**Scope and Limitation**

The program is intended for users who are concerned for their health in terms of their calorie intake that they eat every single day. This is also intended for users who have a goal that they want to achieve in their diet. The program is intended for users in the right age of usage in which are mainly in the age range of 13-59 years old who are indulged in achieving a diet by determining the calorie intake for every day and computing the BMI of each user. The program only includes 33 types of foods in which ranging from fruits, vegetables, meat, dairy, and other products that are commonly intake by users. This way, users can know what are the most common foods that each user or person eat every day. The functions of the program only include the following: adding food to the database of your food intake, removing the food from the food intake database, updating the user’s account information, searching a specific type of food, viewing all the foods, viewing all the past food intakes of the user, and computing for the BMI of the user and determining which category does the user fall in based on the BMI. The information about the foods only includes the Name, Description, the amount per serving, and the number of calories of the specific food. The information about the user only includes the Name, Gender, Age, Height, Weight, the username and password. The program does not include the different exercising routines that users can do in order to lower their weight. It does not include other foods that are not in the database of the program. The main focus and purpose of the program is to determine the calorie intake of each food and how users will be able to adapt and change the way they intake food. This program also helps users in determining their BMI, which is based on their weight and height.

**User Manual**

1. Getting Started

1.1 Open the program

1.2 The program logo will be shown, find the “Let’s Get Started” button, then click

it.

1.3 It will show the next page of the program and a message welcoming the user. Click “OK” to continue.

2. Create an Account

2.1 The second page is the sign in page. If the user is new, find the “Sign up button” then click it.

2.2 A new page will show, asking for the information of the user to create a new account.

2.3 Fill up the needed information on the space provided.

2.4 Click the confirm button to confirm the creation of new account.

2.5 A message will pop up about the creation of the account. Click “OK” to continue.

3. Signing In

3.1 When already has an account, simply fill up the space provided in the sign in page.

3.2 Click the “Log In” button when finished.

4. Setting a Calorie Goal

4.1 On the main page, find the “Set Goal” button on the middle top of the page, then click it.

4.2 An input box will show, simply type the number of calories you want to consume for the day.

4.3 Click “OK”.

5. Adding Food/s

5.1 Navigate through the list of food by scrolling up and down or find specific food by the “Search By” Drop down list, then typing about the food depending on your chosen way of search in the “Search” Text box provided.

5.2 Click the food from the list the you want to add.

5.3 Click the “Add” Button and it will be shown to the list of added foods below.

6. Removing Food/s

6.1 Remove an already added food by clicking the food from the list of added foods below the list of foods.

6.2 Click the “Remove” button.

6.3 A message box will appear confirming the removal of the added food. Click “Yes” to confirm.

7. Confirming Items

7.1 When there are food on the added food list, click the “Confirm Items” button to complete the list of food that is consumed or will be consumed.

7.2 A message box will appear, confirming the list of items will be saved. Click “Yes” to Confirm.

7.3 Another message box will appear indicating that the list of food are saved. Click “OK”, then the total calories of the added items will be subtracted to the calorie goal.

8. Getting an Idea on What to Put on the Calorie Goal

8.1 Go to the “Calculator” tab by clicking the text calculator above.

8.2 The information of the user will be automatically inputted there, if it needs changes, just change the information on the boxes provided.

8.3 Click the “Calculate” Button to show the possible calorie goal along with your Body Mass Index (BMI).

9. Checking the History of the Added List of Foods Foods

9.1 Go to the “History” tab by clicking the text History above.

9.2 The Added Time and Date can be seen here, also the list of foods added and the total calories.

10. Deleting History

10.1 On the “History” tab, click the set of food added in the list.

10.2. Click the “Delete” Button to delete specifically.

10.3 A message box will be shown confirming the deletion. Click “Yes” to confirm.

10.4 Click the “Delete All History” button to delete all data on the history.

10.5 A message box will be shown confirming the deletion. Click “Yes” to confirm.

11. Change Account Information

11.1 Go to the “Account Settings” tab by click the text Account Settings above.

11.2 Simply change the information by typing on the text boxes that already has written information.

11.3 When done, Click the “Update Account” button.

11.4 A message box will appear about confirming the changes that are made on the account information. Click “Yes” to confirm.

11.5 Another message box will appear about successfully updating the account information. Click “OK” to confirm.

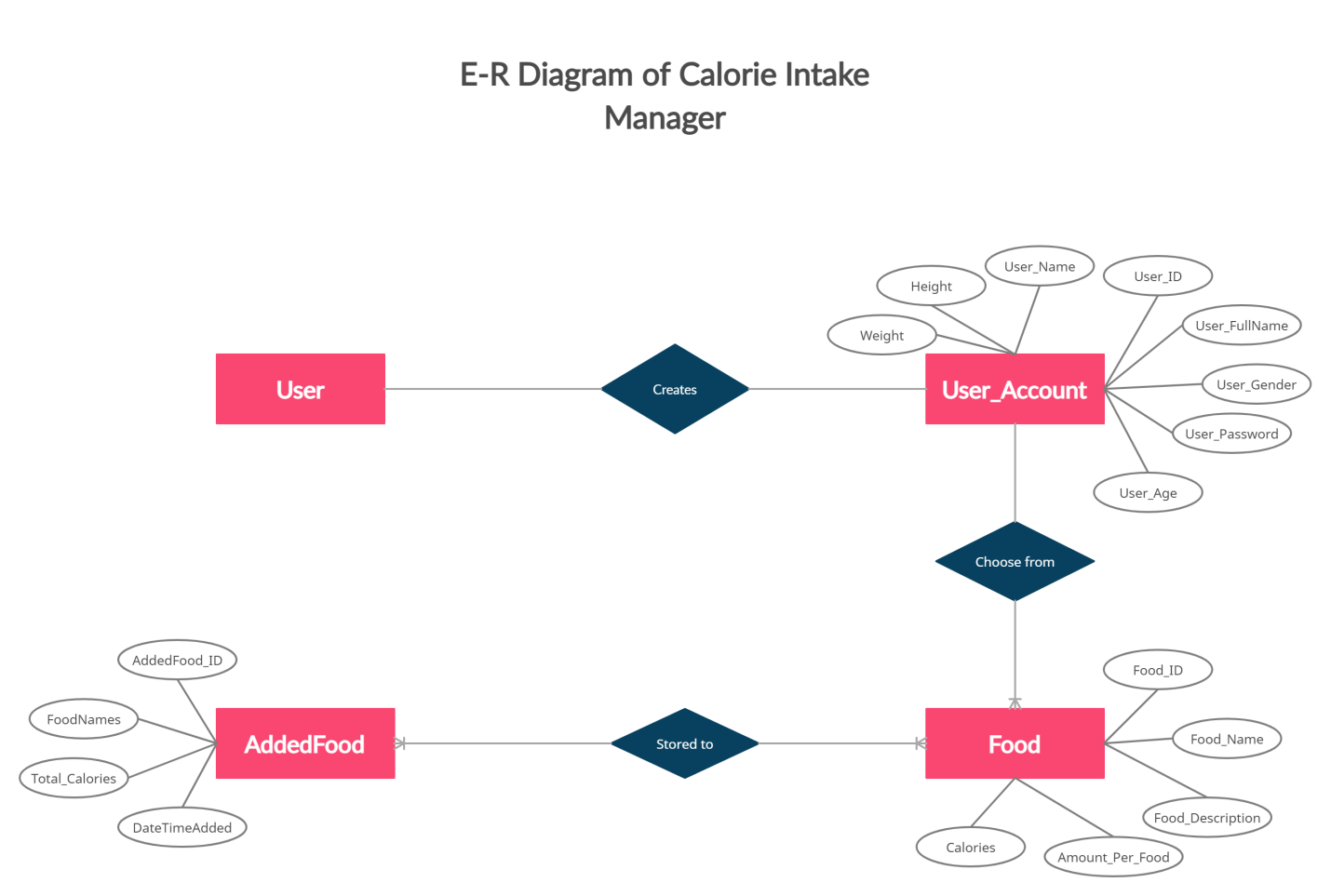
12. Logging Out the Account

12.1 Click the “Log Out” button on the top right corner.

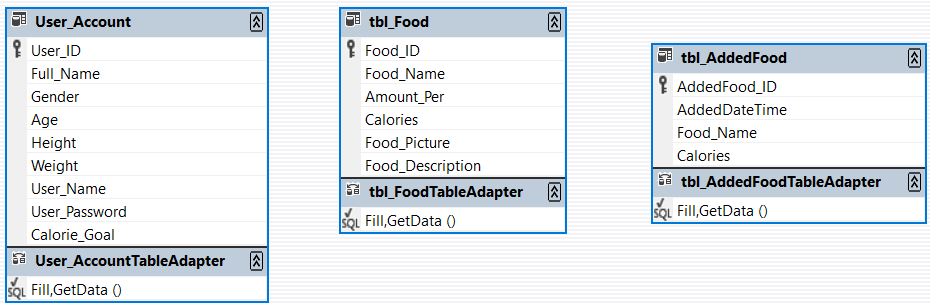
12.2 A message box will appear about the confirmation of logging out of the account. Click “Yes” to confirm.

12.3 The account will be logged out. A message box will appear saying thank you to the user.

**Entity Relationship Diagram**

****

**Class Diagram**

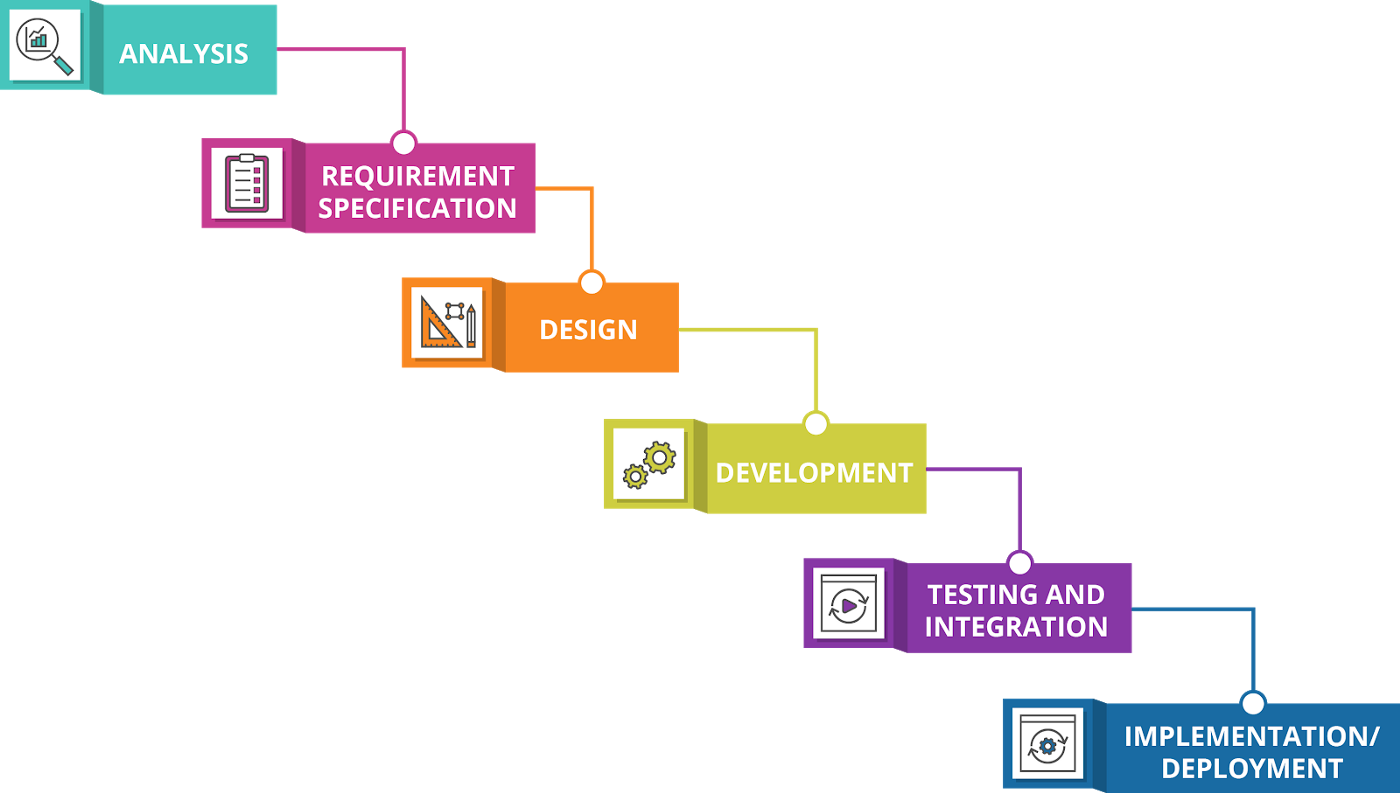
****

**Software Development Life Cycle**

* Communication – we created a survey form in order to determine the needs of the users in the program
* Requirement gathering – We collect and gather requirements based on the results of the survey form and also based on the researched contents we have done
* Feasibility study – based on the survey that we have done; we can conclude that our program will be feasible and useful in the future because the results says that most of the respondents says that they would use an application in order to calculate the calorie intake of the users.
* System analysis – we analyze the program in which is based on its content. We analyze what type of content and what different approaches in coding do we integrate and we use the coding methods of interface, inheritance, adding, deleting and updating.
* Software design – the design that we finalize is the theme of green and white so that it is connected and in line with our program.
* Coding – the different methods and approaches that we use in coding this program are we include and use the methods of interface, inheritance, adding, deleting and updating.
* Testing – we tested the program, and we discover minor errors and we fix them immediately and check again for other errors while testing. If there are no other errors, we move on to the next step.
* Integration - When the program is done and error free, we integrate by implementing other methods and steps like, compatibility for smartphone users and apple users.
* Implementation - This is when we will publish the application that we made to platforms like apple play and play store.
* Operation and Maintenance - When the program is already out in the world, we will do maintenance to keep its relevance, and fix problems if there’s any.
* Disposition - When the program has lost its relevance, we will terminate it and delete the program.

**Software Model**

Waterfall Model

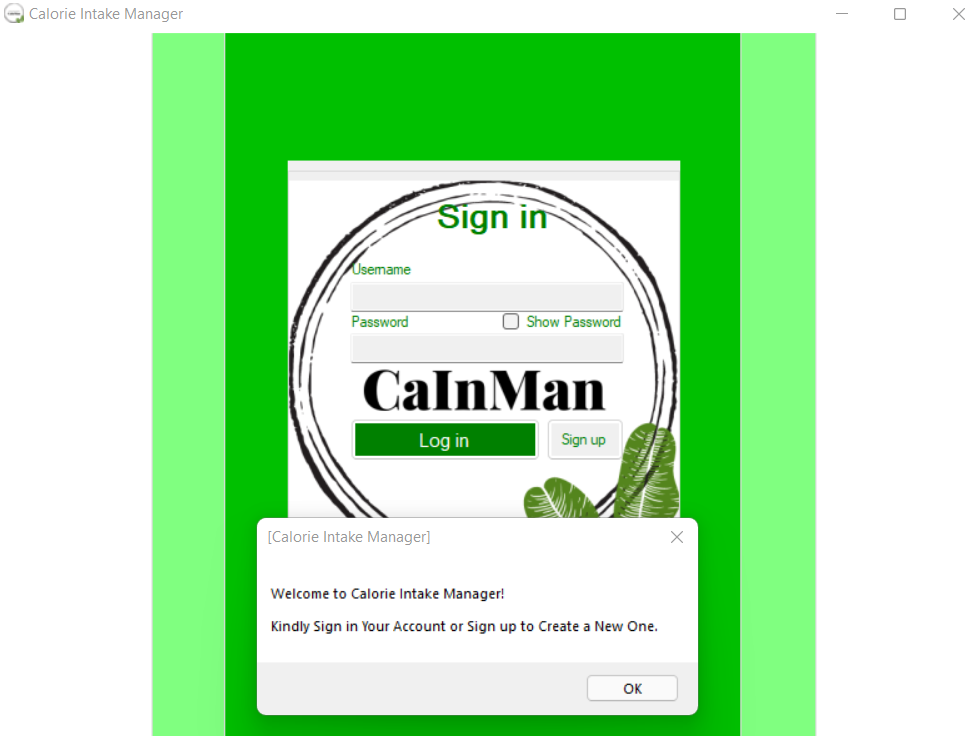
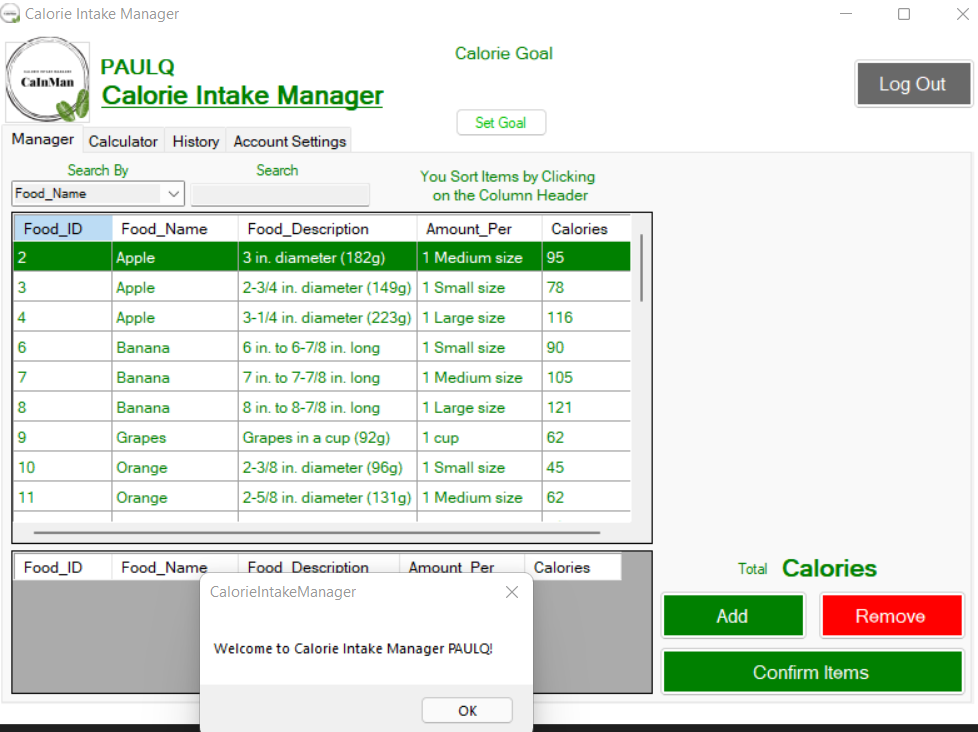
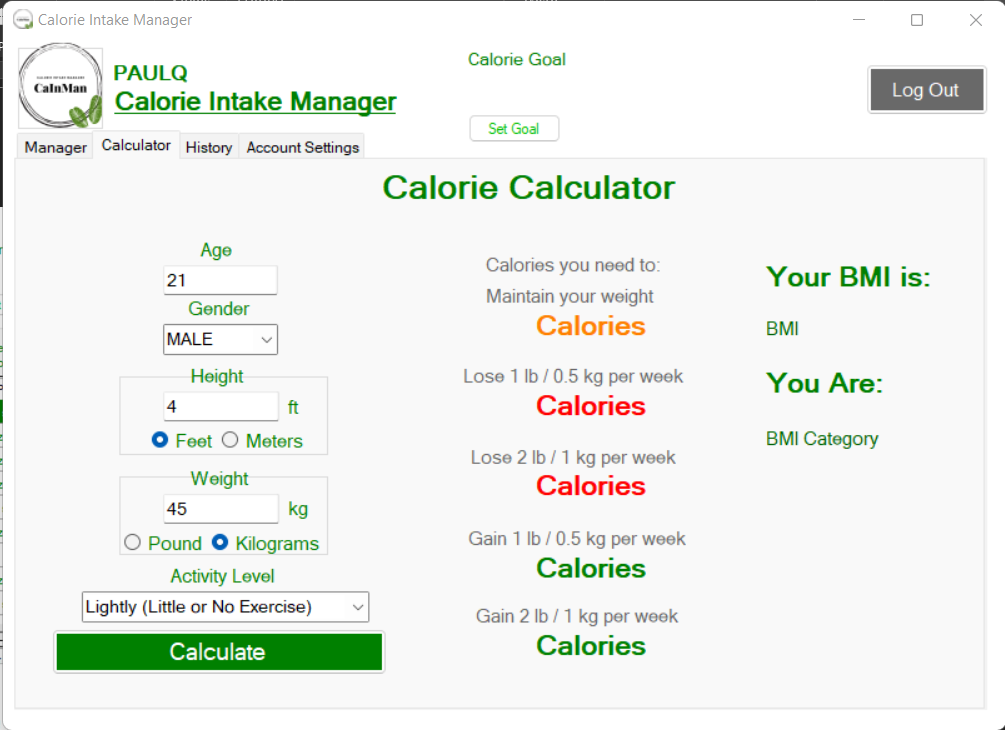
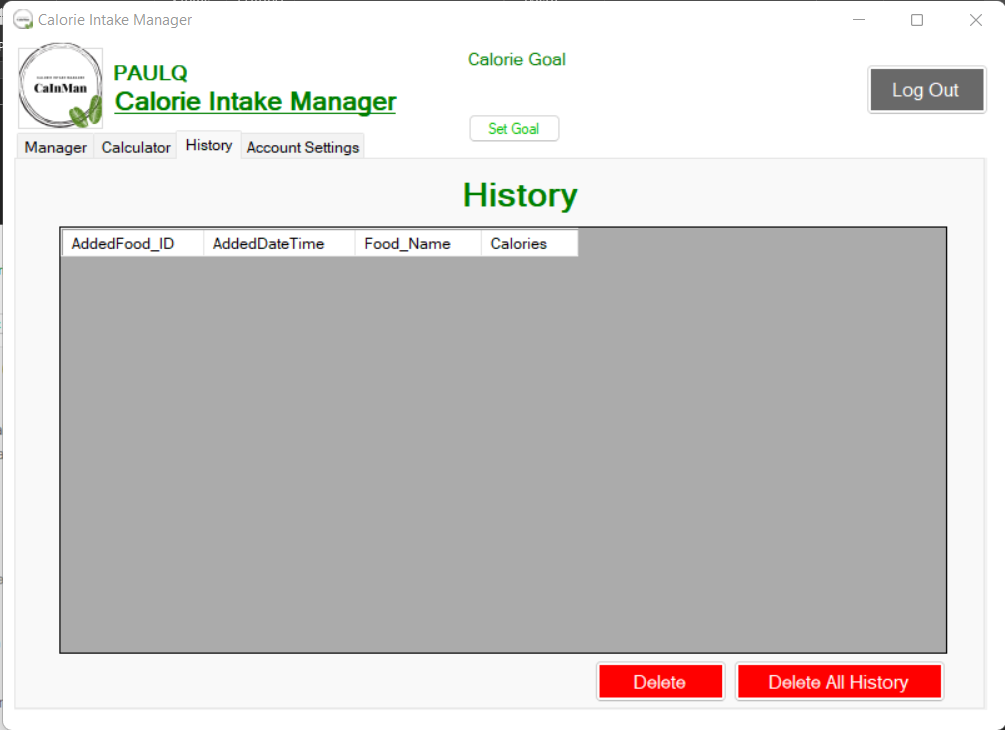
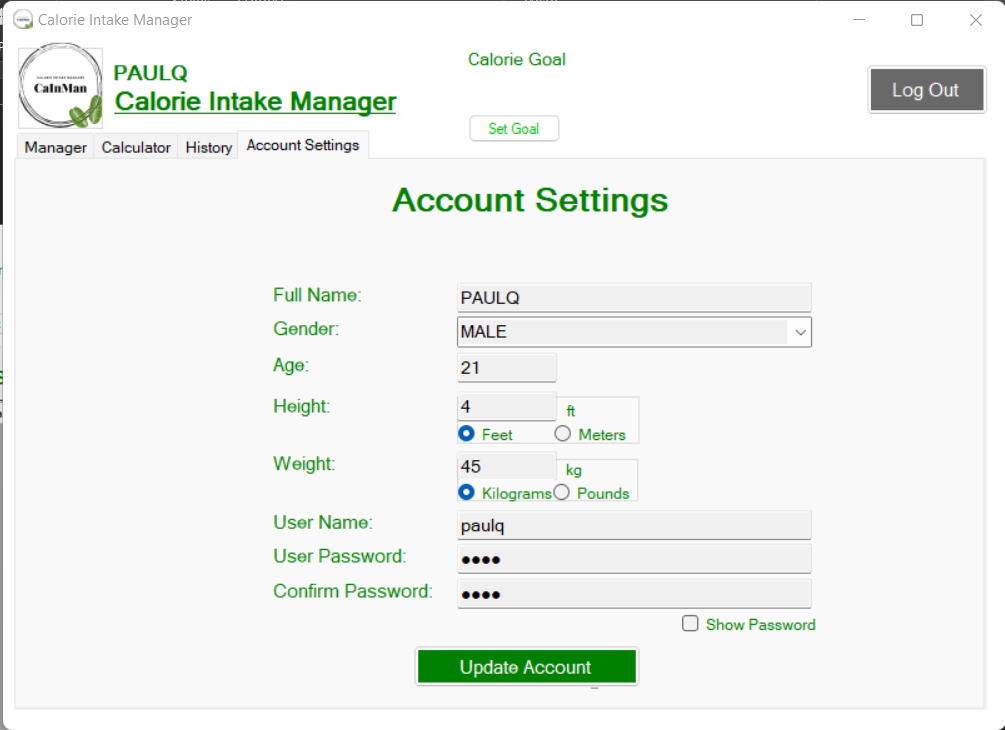


We chose the waterfall model because our program is a simple, and it is a short program. The requirements of our program do not change constantly, and it is not changing regularly because in our program, the number of calories that each specific type of food has does not change regularly. It is a constant amount throughout the course of time. All the requirements in order to create this program is easily accessible and available, and it is not changing regularly. All the resources that are included in the program are already available, and we will implement and use it in our program in which is a success. The waterfall model is best for our program because it is only a short program, all the requirements and resources that are needed in order to create the program are already available and the information that are included in the program, like the amount of calorie intake per food, does not change over the course of time.

**Flow Chart**



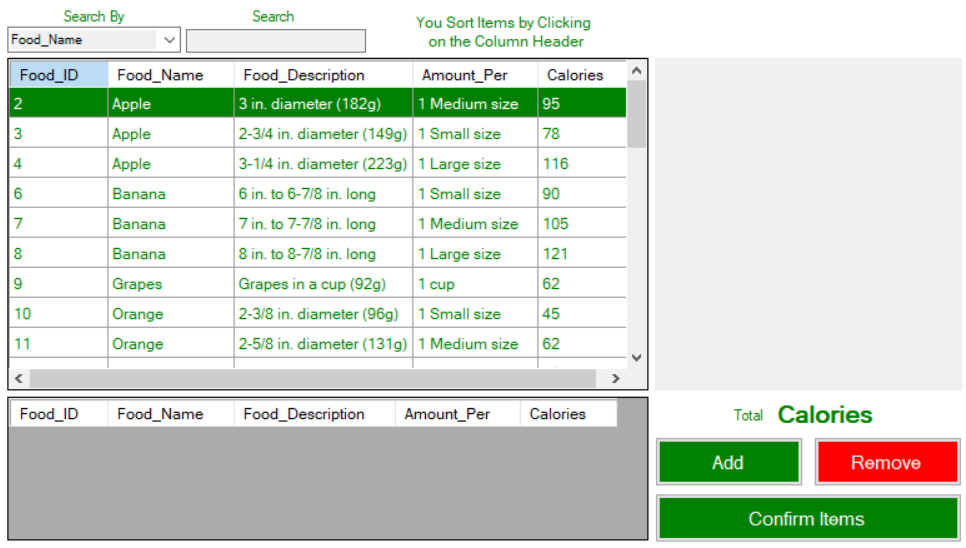
**Process Flow Chart**

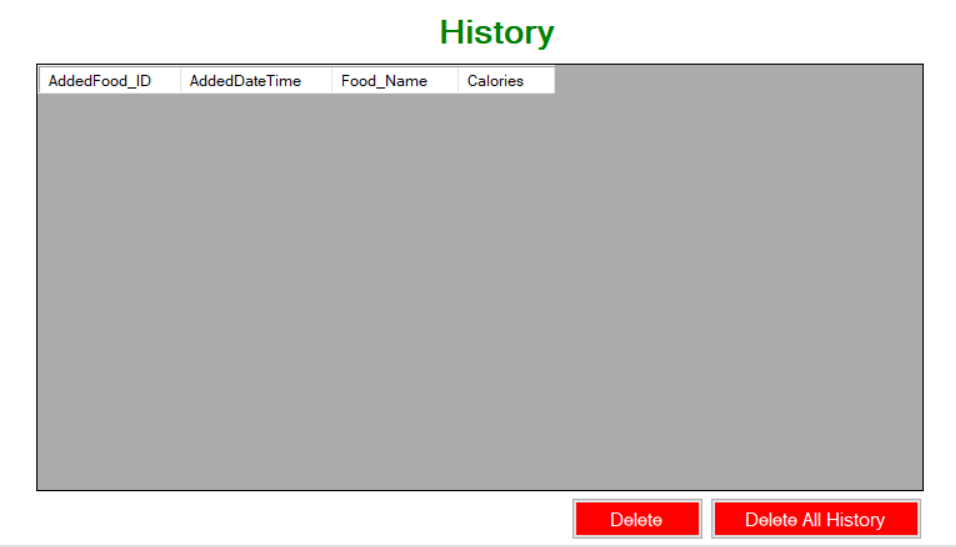
**** ****    

**Iterations of the Program**

* **Initial Program**



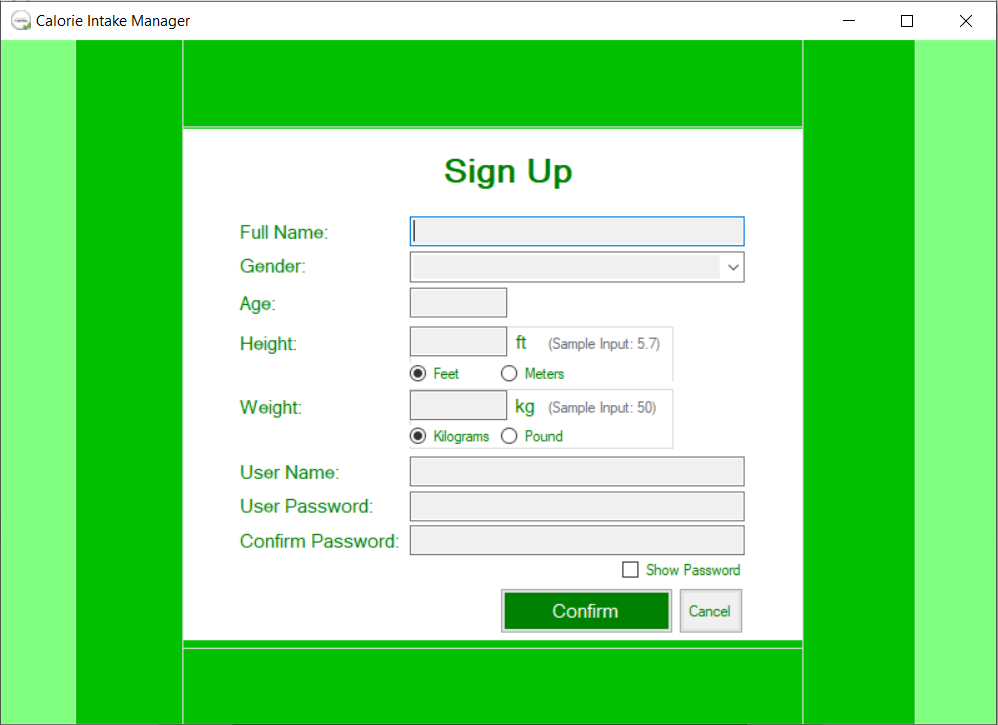




The Initial program consists only of the Logo Page, the Main Page which is the calorie intake manager, and the History Page. The program works as it has its essential purpose which is to count calories based on the foods that are available in the program. It stores the user data, and show it in the history page. We though that it can work as it is but we realized that it is limited to one user per device, and the data in the history will be gone once the app is deleted from the device. A user cannot also access their data from another device besides from where they started using it.

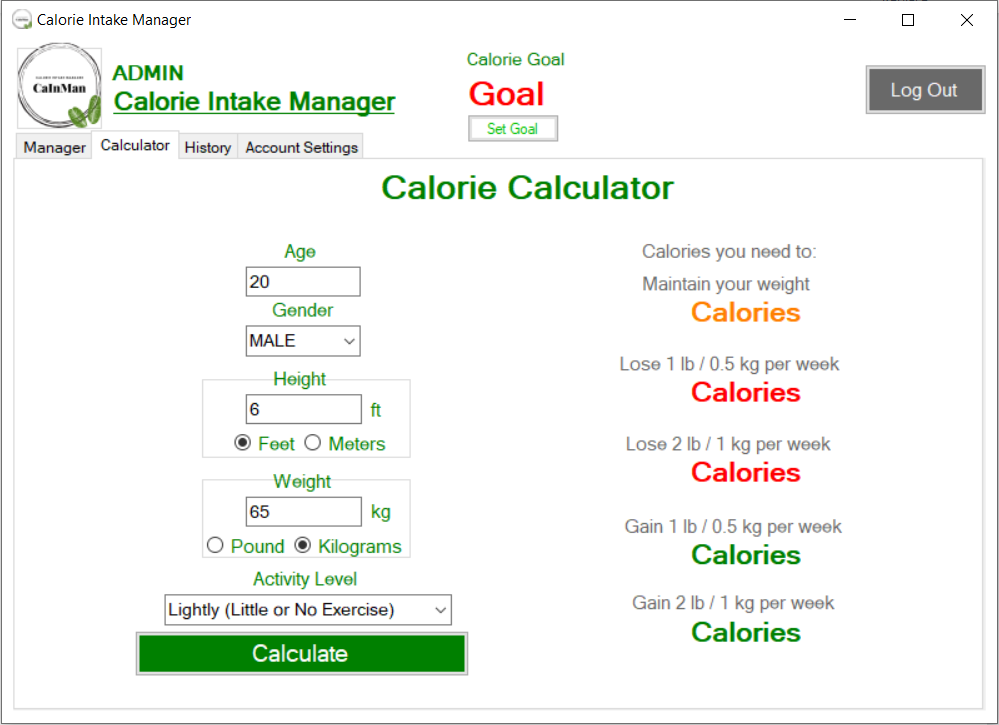
**1st Iteration**





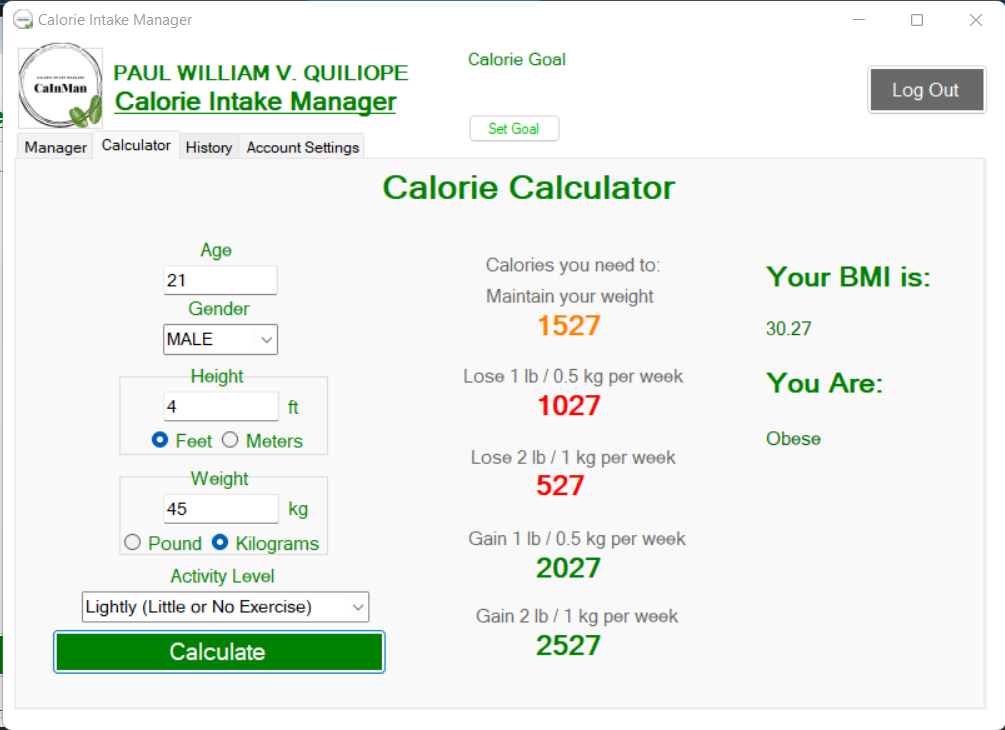
We resolved the problem with the limited access of users to their data by adding accounts to the program. Now, users can have their own personal account. All they need to do is to create one in a few easy steps, then it’s done. Users can now also access their data using their own accounts to different devices. After that, we realized that we are asking for their height and weight, that we should also make use of that information. Something that will be useful to them, something that they can use as their first move or start, especially when they are new to their diet. Something that can also motivate them to continue pursuing their goal.

**2nd Iteration**

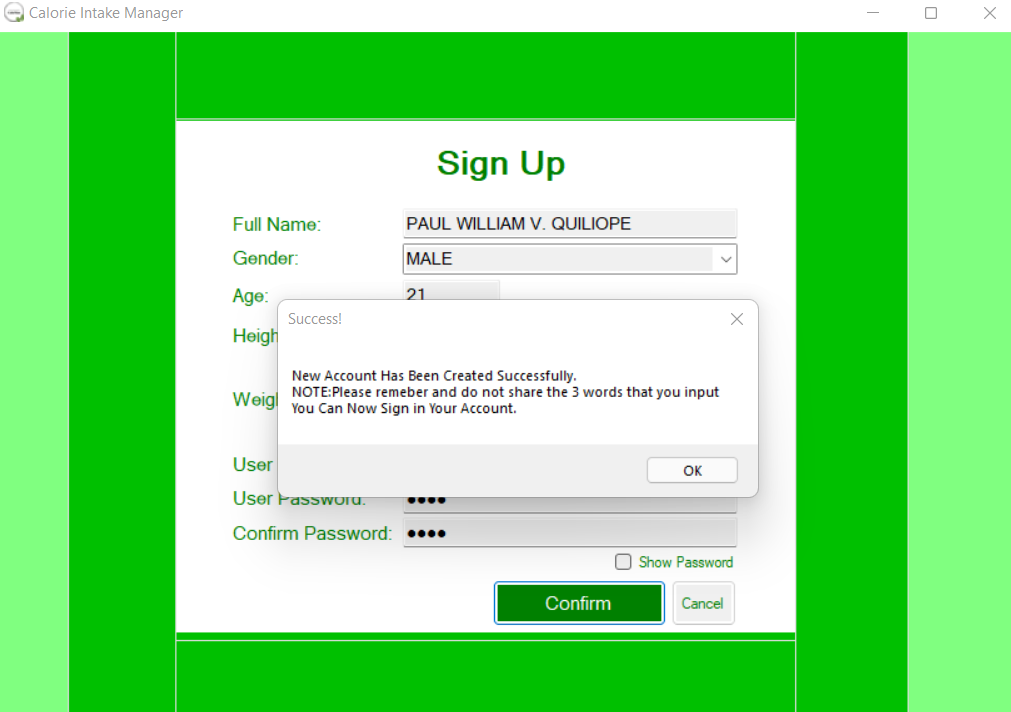
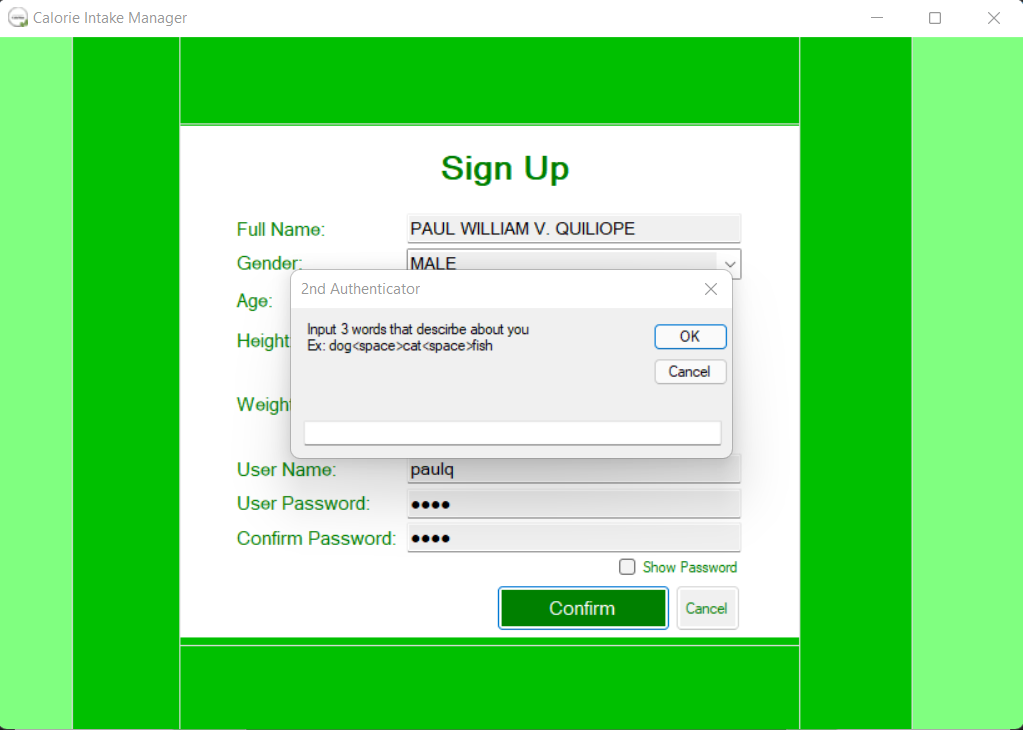


To compensate for what we have realized on the first iteration, we added to the program a calculator. This Calculator counts the number of calories that the user needs to consume per day in order to either maintain, gain or lose weight. It makes use of the information of the user on their age, gender, height, and weight. We also added a Calorie Goal per day that the user can set to keep better track of the number of calories that they will need to consume.

**3rd Iteration**

****

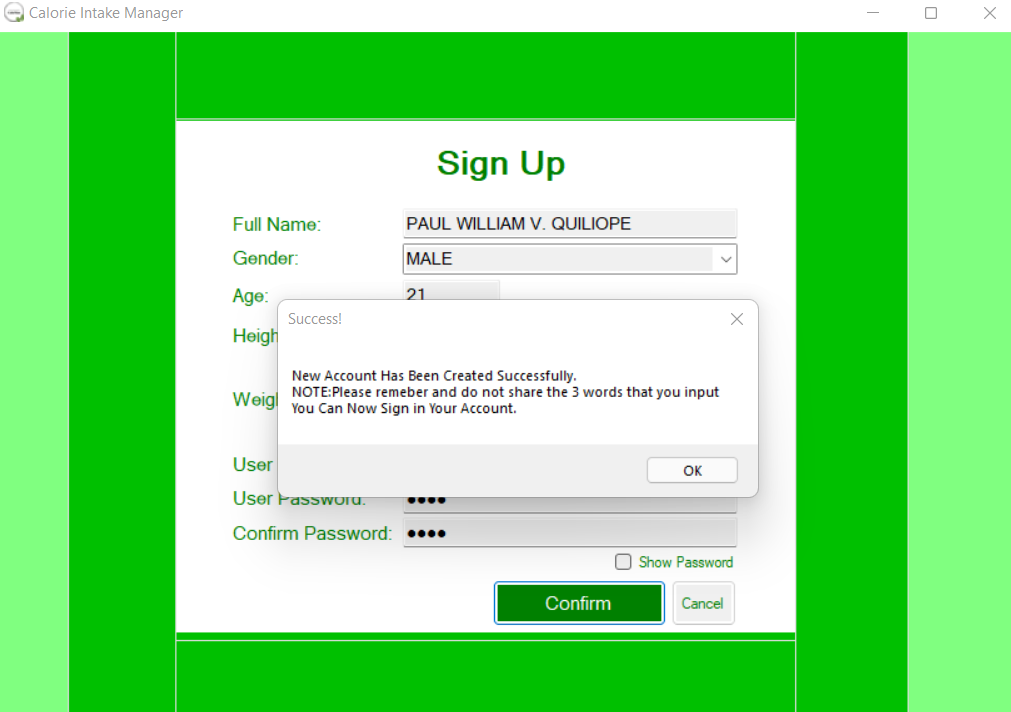
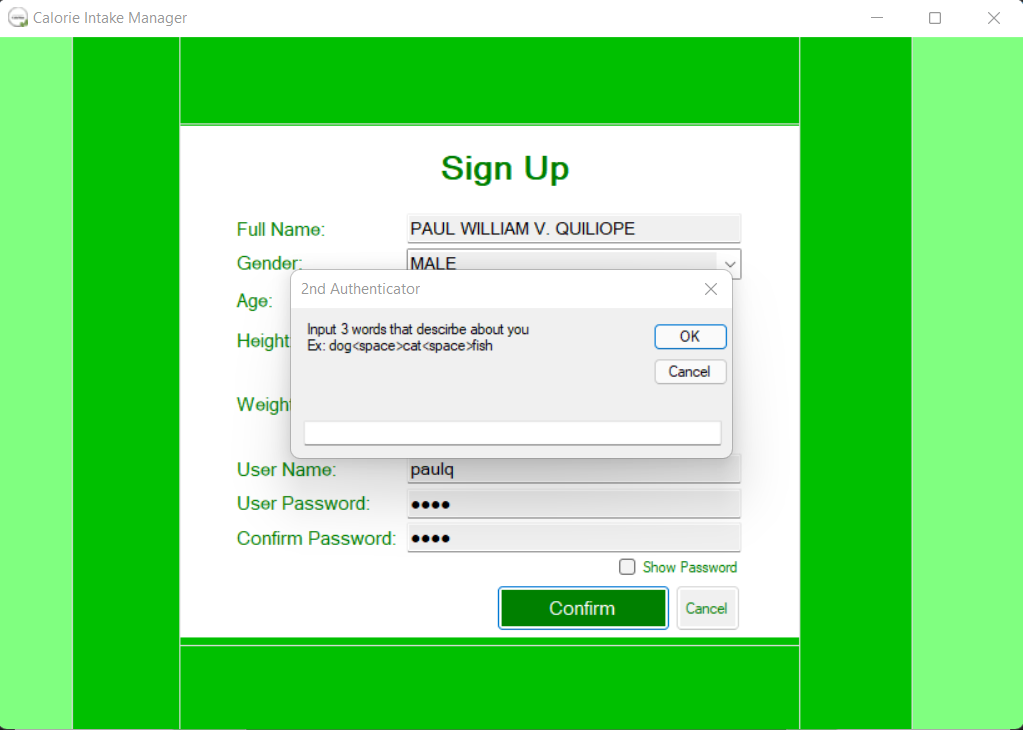
In order to add more functions and content in our program, we added with the calorie calculator the BMI calculator where in it can determine you BMI by computing and using your height and weight in order to determine the BMI of the user. Also, in accordance with this, we also added the categories that fall on each BMI that the system computed. The reason for this is for users to determine in which category would they fall in. This way users can know and change their lifestyle if they saw the BMI calculation and its corresponding category.

**4th Iteration**

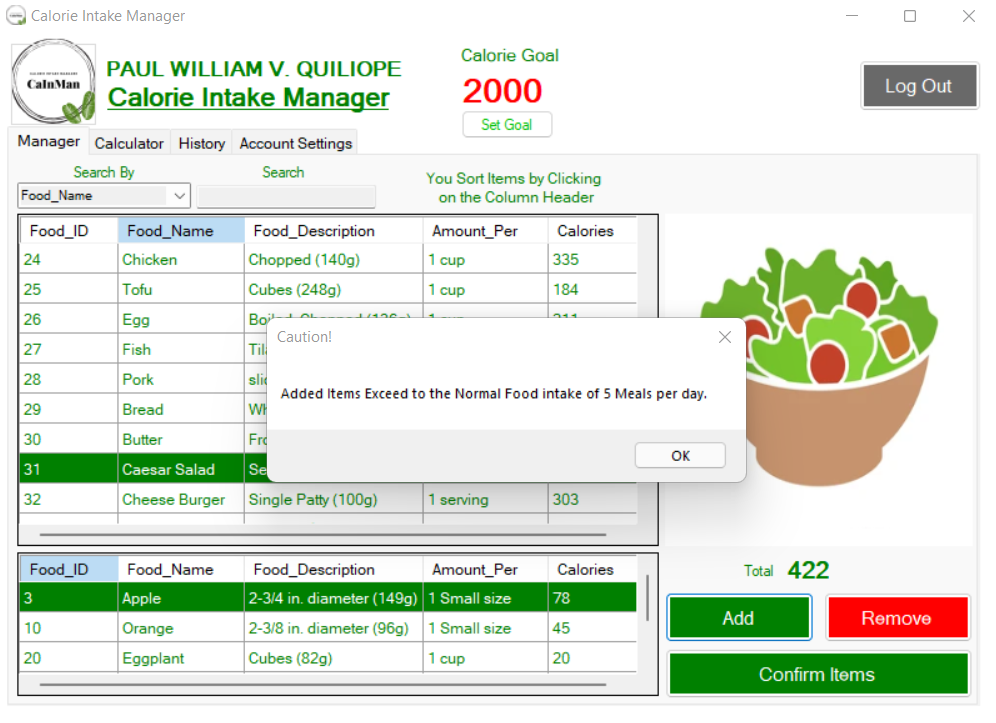
The last iteration that we did in our program is in par with the suggestion that the panel suggest. And that is to heighten and to secure the security of each user with their own respective accounts. With this we add and develop this feature in our program where in we add a 2-factor authentication every time the user logs in into their own account. The second authentication will appear when the user creates a new account and when the user correctly inputs its username and password. The second authentication will ask the user to input three respective words that describes them. If successful, the program will notify that they should remember the words and the user should not share the 3-word authentication that they have inputted as it will make their account more secure.

**Suggestions in the Program**

* For Security – we added a 2nd-Authentication in which the user inputs 3 words

****

* For limit in the added foods – we added a message box when the user exceeds in choosing and adding the foods in which is 5 foods per transaction



* For the foods that can be added – we added more types of foods to the database



**Program Codes**

**WelcomeForm.vb**

Public Class WelcomeForm

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

SignInForm.Show()

MsgBox("Welcome to Calorie Intake Manager!" & vbNewLine & vbNewLine & "Kindly Sign in Your Account or Sign up to Create a New One.", MsgBoxStyle.OkOnly, "[Calorie Intake Manager]")

Me.Hide()

End Sub

End Class

**SignInForm.vb**

Public Class SignInForm

Private MyStoredProcedures As New UserAccountStoredProceduresDataContext

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles btnLogin.Click

User\_AccountBindingSource.DataSource = MyStoredProcedures.checkUser(txtName.Text, txtPass.Text)

If User\_AccountBindingSource.Count = 0 Then

MsgBox("There is no existing user account yet!")

ElseIf txtName.Text Like User\_NameTextBox.Text = True AndAlso txtPass.Text Like User\_PasswordTextBox.Text = True Then

Me.Hide()

CIMMainForm.Show()

CIMMainForm.lblName.Text = Full\_NameTextBox.Text

MsgBox("Welcome to Calorie Intake Manager " & Full\_NameTextBox.Text & "!")

txtName.Text = ""

txtPass.Text = ""

Else

MsgBox("Username and Password Did Not Match Any Existing Data.", MsgBoxStyle.OkOnly, "Invalid!")

End If

End Sub

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

New\_Account.Show()

End Sub

Private Sub User\_AccountBindingNavigatorSaveItem\_Click(sender As Object, e As EventArgs)

Me.Validate()

Me.User\_AccountBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.UserAccountDataSet)

End Sub

Private Sub User\_AccountBindingNavigatorSaveItem\_Click\_1(sender As Object, e As EventArgs) Handles User\_AccountBindingNavigatorSaveItem.Click

Me.Validate()

Me.User\_AccountBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.UserAccountDataSet)

End Sub

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

'TODO: This line of code loads data into the 'UserAccountDataSet.User\_Account' table. You can move, or remove it, as needed.

'Me.User\_AccountTableAdapter.Fill(Me.UserAccountDataSet.User\_Account)

End Sub

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

If chShowPass.Checked Then

txtPass.UseSystemPasswordChar = False

Else

txtPass.UseSystemPasswordChar = True

End If

End Sub

End Class

**New\_Account.vb**

Public Class New\_Account

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

Me.Validate()

Me.User\_AccountBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.UserAccountDataSet)

End Sub

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

'TODO: This line of code loads data into the 'UserAccountDataSet.User\_Account' table. You can move, or remove it, as needed.

Me.User\_AccountTableAdapter.Fill(Me.UserAccountDataSet.User\_Account)

User\_AccountBindingSource.AddNew()

End Sub

Private MyStoredProcedures As New UserAccountStoredProceduresDataContext

Dim account As New AccountClass

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

account.getName = Full\_NameTextBox.Text

account.getGender = GenderComboBox.Text

account.getAge = Val(AgeTextBox.Text)

account.getHeight = Val(HeightTextBox.Text)

account.getWeight = Val(WeightTextBox.Text)

If (Full\_NameTextBox.Text = "" AndAlso AgeTextBox.Text = "" AndAlso HeightTextBox.Text = "" AndAlso User\_NameTextBox.Text = "" AndAlso txtConfirmPass.Text AndAlso User\_PasswordTextBox.Text = "") = True Then

MsgBox("Please Complete All Missing Boxes.", MsgBoxStyle.OkOnly, "Unsuccessful!")

ElseIf account.getAge = -1 Then

MsgBox("You're Too Young to Use This Application!", MsgBoxStyle.OkOnly, "Invalid!")

ElseIf account.getAge = 0 Then

MsgBox("You're Too Old to Use This Application!", MsgBoxStyle.OkOnly, "Invalid!")

ElseIf account.getHeight = 0 Then

MsgBox("Height Input is Either Too Low or Too High." & vbNewLine & "Please Check Again!", MsgBoxStyle.OkOnly, "Invalid!")

ElseIf account.getWeight = 0 Then

MsgBox("Weight is too low.", MsgBoxStyle.OkOnly, "Invalid!")

Else

If User\_PasswordTextBox.Text Like txtConfirmPass.Text = True Then

MyStoredProcedures.CreateNewAccount(account.getName, account.getGender, account.getAge, account.getHeight, account.getWeight, User\_NameTextBox.Text, User\_PasswordTextBox.Text)

MsgBox("New Account Has Been Created Successfully." & vbNewLine & "You Can Now Sign in Your Account.", MsgBoxStyle.OkOnly, "Success!")

Me.Close()

Else

MsgBox("User Password and Confirm Password are not the same." & vbNewLine & "Please Try Again.", MsgBoxStyle.OkOnly, "Unsuccessful!")

End If

End If

End Sub

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

User\_AccountBindingSource.RemoveCurrent()

Me.Close()

End Sub

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

If chShowPass.Checked Then

User\_PasswordTextBox.UseSystemPasswordChar = False

txtConfirmPass.UseSystemPasswordChar = False

Else

User\_PasswordTextBox.UseSystemPasswordChar = True

txtConfirmPass.UseSystemPasswordChar = True

End If

End Sub

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

If rdoMeters.Checked Then

lblHeight.Text = "m"

lblSH.Text = "(Sample Input: 1.74)"

account.getHeight = Val(HeightTextBox.Text)

HeightTextBox.Text = account.Divide(account.getHeight, 3.281)

Else

lblHeight.Text = "ft"

lblSH.Text = "(Sample Input: 5.7)"

account.getHeight = Val(HeightTextBox.Text)

HeightTextBox.Text = account.Multiply(account.getHeight, 3.281)

End If

End Sub

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

If rdoPound.Checked Then

lblWeight.Text = "lb"

lblSW.Text = "(Sample Input: 110)"

account.getWeight = Val(WeightTextBox.Text)

WeightTextBox.Text = account.Multiply(account.getWeight, 2.205)

Else

lblWeight.Text = "kg"

lblSW.Text = "(Sample Input: 50)"

account.getWeight = Val(WeightTextBox.Text)

WeightTextBox.Text = account.Divide(account.getWeight, 2.205)

End If

End Sub

End Class

**CIMMainForm.vb**

**Public Class CIMMainForm**

**Private MyStoredProcedures As New UserAccountStoredProceduresDataContext**

**Dim Food As New FoodClass**

**Dim Account As New AccountClass**

**Dim totalCal As Integer**

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

**DataGridView1.Rows.Add(txtID.Text, txtName.Text, txtDescription.Text, txtAmount.Text, txtCalories.Text)**

**totalCal += Val(txtCalories.Text)**

**lblTotalCal.Text = totalCal**

**End Sub**

**Private Sub Tbl\_FoodBindingNavigatorSaveItem\_Click(sender As Object, e As EventArgs) Handles Tbl\_FoodBindingNavigatorSaveItem.Click**

**Me.Validate()**

**Me.Tbl\_FoodBindingSource.EndEdit()**

**Me.TableAdapterManager.UpdateAll(Me.UserAccountDataSet)**

**End Sub**

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

**'TODO: This line of code loads data into the 'UserAccountDataSet.User\_Account' table. You can move, or remove it, as needed.**

**Me.User\_AccountTableAdapter.Fill(Me.UserAccountDataSet.User\_Account)**

**'TODO: This line of code loads data into the 'UserAccountDataSet.tbl\_AddedFood' table. You can move, or remove it, as needed.**

**Me.Tbl\_AddedFoodTableAdapter.Fill(Me.UserAccountDataSet.tbl\_AddedFood)**

**'TODO: This line of code loads data into the 'UserAccountDataSet.tbl\_Food' table. You can move, or remove it, as needed.**

**Me.Tbl\_FoodTableAdapter.Fill(Me.UserAccountDataSet.tbl\_Food)**

**User\_AccountBindingSource.DataSource = User\_AccountBindingSource.Current**

**Food.getHeight = Val(HeightTextBox1.Text)**

**If Food.getHeight = -1 Then**

**rdoMeters.Checked = True**

**rdoM.Checked = True**

**ElseIf Food.getHeight = 0 Then**

**rdoFeet.Checked = True**

**rdoFT.Checked = True**

**End If**

**Food.getWeight = Val(WeightTextBox1.Text)**

**If Food.getWeight = 0 Then**

**rdoKilograms.Checked = True**

**rdoKG.Checked = True**

**ElseIf Food.getWeight = -1 Then**

**rdoPound.Checked = True**

**rdoLB.Checked = True**

**End If**

**End Sub**

**Private Sub txtSearch\_TextChanged(sender As Object, e As EventArgs) Handles txtSearch.TextChanged**

**If cmbSearchBy.Text = "Food\_Name" Then**

**Tbl\_FoodBindingSource.DataSource = MyStoredProcedures.SelectByFoodName(txtSearch.Text)**

**ElseIf cmbSearchBy.Text = "Food\_ID" Then**

**Food.foodSearchID = Val(txtSearch.Text)**

**Tbl\_FoodBindingSource.DataSource = MyStoredProcedures.SelectByFoodID(Food.foodSearchID)**

**ElseIf cmbSearchBy.Text = "Description" Then**

**Tbl\_FoodBindingSource.DataSource = MyStoredProcedures.SelectByDescription(txtSearch.Text)**

**ElseIf cmbSearchBy.Text = "Amount\_Per" Then**

**Tbl\_FoodBindingSource.DataSource = MyStoredProcedures.SelectByAmountPer(txtSearch.Text)**

**End If**

**End Sub**

**Private Sub cmbSearchBy\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cmbSearchBy.SelectedIndexChanged**

**Dim starts, ends As Integer**

**Dim ask As MessageBoxOptions**

**If cmbSearchBy.Text = "Calories" Then**

**Do Until ask = vbCancel Or ends > 0**

**Try**

**starts = InputBox("Input Calories Starting From:")**

**ends = InputBox("To:")**

**Tbl\_FoodBindingSource.DataSource = MyStoredProcedures.SelectByCalories(starts, ends)**

**Catch ex As Exception**

**ask = MsgBox("Invalid Input!" & vbNewLine & "Do You Want to Try Again?", MsgBoxStyle.OkCancel, "Error!")**

**End Try**

**Loop**

**End If**

**End Sub**

**Private Sub Tbl\_FoodDataGridView\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles Tbl\_FoodDataGridView.CellClick**

**Dim selectedRowIndex As Integer**

**selectedRowIndex = e.RowIndex**

**Dim row As New DataGridViewRow**

**If selectedRowIndex >= 0 Then**

**row = Tbl\_FoodDataGridView.Rows(selectedRowIndex)**

**txtID.Text = row.Cells(0).Value.ToString()**

**txtName.Text = row.Cells(1).Value.ToString()**

**If txtName.Text = "Banana" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/vector-cartoon-banana-260nw-154201229.jpg"**

**ElseIf txtName.Text = "Apple" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/red-apple-vector-illustration-600w-307260020.jpg"**

**ElseIf txtName.Text = "Grapes" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/bunch-grapes-on-white-background-600w-254889667.jpg"**

**ElseIf txtName.Text = "Orange" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/oranges-green-leaves-illustration-600w-346152509.jpg"**

**ElseIf txtName.Text = "Pear" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/green-pear-leaf-illustration-600w-349819007.jpg"**

**ElseIf txtName.Text = "Peach" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/fresh-peach-slice-illustration-600w-501544411.jpg"**

**ElseIf txtName.Text = "Pineapple" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/pineapple-icon-isolated-on-white-600w-538361251.jpg"**

**ElseIf txtName.Text = "Strawberry" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/garden-strawberry-icon-vector-illustration-600w-1206914827.jpg"**

**ElseIf txtName.Text = "Watermelon" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/illustration-watermelon-slice-concept-healthy-600w-1312712918.jpg"**

**ElseIf txtName.Text = "Asparagus" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/asparagus-vector-illustration-sparrow-grass-600w-2083088416.jpg"**

**ElseIf txtName.Text = "Broccoli" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/broccoli-600w-452774542.jpg"**

**ElseIf txtName.Text = "Carrot" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/carrot-clipart-graphic-600w-1312564031.jpg"**

**ElseIf txtName.Text = "Eggplant" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/eggplant-emoji-icon-cartoon-drawing-600w-1431018887.jpg"**

**ElseIf txtName.Text = "Lettuce" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/green-lettuce-on-white-background-600w-1594530688.jpg"**

**ElseIf txtName.Text = "Tomato" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/tomato-isolated-single-simple-cartoon-600w-297687881.jpg"**

**ElseIf txtName.Text = "Beef" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/tenderloin-steak-600w-102555731.jpg"**

**ElseIf txtName.Text = "Chicken" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/chicken-thighs-icon-isolated-legs-600w-602517836.jpg"**

**ElseIf txtName.Text = "Tofu" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/simple-cute-illustration-chilled-tofu-600w-1951698601.jpg"**

**ElseIf txtName.Text = "Egg" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/vector-food-icon-chicken-boiled-600w-1288868188.jpg"**

**ElseIf txtName.Text = "Fish" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/fish-600w-101438332.jpg"**

**ElseIf txtName.Text = "Pork" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/pork-meat-piece-bone-cut-600w-2067870482.jpg"**

**ElseIf txtName.Text = "Bread" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/illustration-bread-slices-on-white-600w-113331841.jpg"**

**ElseIf txtName.Text = "Butter" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/sliced-margarine-block-baking-ingredient-600w-505798189.jpg"**

**ElseIf txtName.Text = "Caesar Salad" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/garden-salad-lettuce-tomatoes-bread-600w-363761639.jpg"**

**ElseIf txtName.Text = "Cheese Burger" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/colorful-bright-hamburger-stock-vector-600w-1706527666.jpg"**

**ElseIf txtName.Text = "Pizza" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/pizza-vector-cartoon-clipart-design-600w-1390050710.jpg"**

**ElseIf txtName.Text = "Potato" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/vector-product-icon-vegetable-white-600w-1385664908.jpg"**

**ElseIf txtName.Text = "Rice" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/rice-bowl-600w-452774545.jpg"**

**ElseIf txtName.Text = "Beer" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-illustration/mug-beer-leaking-frothy-high-600w-1822043789.jpg"**

**ElseIf txtName.Text = "Soft Drink" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/cartoon-beverage-drink-design-isolated-600w-486000895.jpg"**

**ElseIf txtName.Text = "Milk" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-vector/tall-glass-milk-carton-box-600w-2017090295.jpg"**

**ElseIf txtName.Text = "Yogurt" Then**

**FoodPic\_PictureBox1.ImageLocation = "https://image.shutterstock.com/image-illustration/freehand-drawn-cartoon-yogurt-600w-455810584.jpg"**

**End If**

**txtDescription.Text = row.Cells(2).Value.ToString()**

**txtAmount.Text = row.Cells(3).Value.ToString()**

**txtCalories.Text = row.Cells(4).Value.ToString()**

**End If**

**End Sub**

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

**Dim foodNames As String**

**Dim calories As Integer**

**For Each row As DataGridViewRow In DataGridView1.Rows**

**foodNames = foodNames & row.Cells(1).Value & " | "**

**calories += row.Cells(4).Value**

**Next**

**If foodNames = "" Then**

**MsgBox("Please Add Items First!", MsgBoxStyle.OkOnly, "Invalid!")**

**Else**

**If Calorie\_GoalLabel1.Text = "" Then**

**Dim ask2 = MsgBox("You can add a Calorie Goal" & vbNewLine & "Do you want To add one first?", MsgBoxStyle.YesNo)**

**If ask2 = MsgBoxResult.No Then**

**MyStoredProcedures.AddFood(foodNames, calories, DateTime.Now)**

**MsgBox("Saved Items Successfully", MsgBoxStyle.OkOnly, "[CalorieIntakeManager] Thank you!")**

**lblTotalCal.Text = ""**

**totalCal = 0**

**DataGridView1.Rows.Clear()**

**Me.Tbl\_AddedFoodTableAdapter.Fill(Me.UserAccountDataSet.tbl\_AddedFood)**

**End If**

**Else**

**Dim ask = MsgBox("Save Items?", MsgBoxStyle.YesNo)**

**If ask = MsgBoxResult.Yes Then**

**MyStoredProcedures.AddFood(foodNames, calories, DateTime.Now)**

**MsgBox("Saved Items Successfully", MsgBoxStyle.OkOnly, "[CalorieIntakeManager] Thank you!")**

**Calorie\_GoalLabel1.Text -= calories**

**If Calorie\_GoalLabel1.Text < 1 Then**

**MsgBox("You've Completed Your Calorie Goal For Today!")**

**Calorie\_GoalTextBox.Text = "Complete"**

**End If**

**lblTotalCal.Text = ""**

**totalCal = 0**

**DataGridView1.Rows.Clear()**

**MyStoredProcedures.AddCalorieGoal(Val(User\_IDTextBox.Text), Full\_NameTextBox.Text, GenderComboBox.Text, Val(AgeTextBox.Text), Val(HeightTextBox.Text), Val(WeightTextBox.Text), User\_NameTextBox.Text, User\_PasswordTextBox.Text, Val(Calorie\_GoalLabel1.Text))**

**Me.Tbl\_AddedFoodTableAdapter.Fill(Me.UserAccountDataSet.tbl\_AddedFood)**

**End If**

**End If**

**End If**

**End Sub**

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

**If DataGridView1.SelectedRows.Count > 0 Then**

**Dim ask = MsgBox("Remove Item?", MsgBoxStyle.YesNo)**

**If ask = MsgBoxResult.Yes Then**

**totalCal -= txtSubtCal.Text**

**lblTotalCal.Text = totalCal**

**DataGridView1.Rows.Remove(DataGridView1.SelectedRows(0))**

**End If**

**Else**

**MsgBox("Please Select Before You Hit Delete!")**

**End If**

**End Sub**

**Private Sub DataGridView1\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView1.CellClick**

**Dim selectedRowIndex As Integer**

**selectedRowIndex = e.RowIndex**

**Dim row As New DataGridViewRow**

**If selectedRowIndex > 0 Then**

**row = DataGridView1.Rows(selectedRowIndex)**

**txtSubtCal.Text = row.Cells(4).Value.ToString()**

**End If**

**End Sub**

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

**Dim goal As Integer**

**Try**

**goal = InputBox("Input Your Calorie Goal")**

**MyStoredProcedures.AddCalorieGoal(Val(User\_IDTextBox.Text), Full\_NameTextBox.Text, GenderComboBox.Text, Val(AgeTextBox.Text), Val(HeightTextBox.Text), Val(WeightTextBox.Text), User\_NameTextBox.Text, User\_PasswordTextBox.Text, goal)**

**Calorie\_GoalLabel1.Text = goal**

**Catch ex As Exception**

**MsgBox("Please Input Numbers Only!")**

**End Try**

**End Sub**

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

**Dim ask = MsgBox("Are You Sure You Want to Log Out?", MsgBoxStyle.YesNo, "[CalorieIntakemanager] Log Out?")**

**If ask = MsgBoxResult.Yes Then**

**Me.Close()**

**SignInForm.Show()**

**MsgBox("Thank you!" & vbNewLine & "Come Again.")**

**End If**

**End Sub**

**Private Sub rdoMeters\_CheckedChanged(sender As Object, e As EventArgs) Handles rdoMeters.CheckedChanged**

**If rdoMeters.Checked Then**

**lblHeight.Text = "m"**

**Account.getHeight = Val(HeightTextBox1.Text)**

**HeightTextBox1.Text = Account.Divide(Account.getHeight, 3.281)**

**Else**

**lblHeight.Text = "ft"**

**Account.getHeight = Val(HeightTextBox1.Text)**

**HeightTextBox1.Text = Account.Multiply(Account.getHeight, 3.281)**

**End If**

**End Sub**

**Private Sub rdoPound\_CheckedChanged(sender As Object, e As EventArgs) Handles rdoPound.CheckedChanged**

**If rdoPound.Checked Then**

**lblWeight.Text = "lb"**

**Account.getWeight = Val(WeightTextBox1.Text)**

**WeightTextBox1.Text = Account.Multiply(Account.getWeight, 2.205)**

**Else**

**lblWeight.Text = "kg"**

**Account.getWeight = Val(WeightTextBox1.Text)**

**WeightTextBox1.Text = Account.Divide(Account.getWeight, 2.205)**

**End If**

**End Sub**

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

**Dim weight, height, weightBMI, heightBMI, BMI As Double**

**Dim activityLevel As Double = 0**

**If rdoFeet.Checked = True Then**

**height = Val(HeightTextBox1.Text) \* 30.48**

**heightBMI = Val(HeightTextBox1.Text) \* 0.3048**

**ElseIf rdoMeters.Checked = True Then**

**height = Val(HeightTextBox1.Text) \* 100**

**heightBMI = Val(HeightTextBox1.Text)**

**End If**

**If rdoKilograms.Checked = True Then**

**weight = Val(WeightTextBox1.Text)**

**weightBMI = Val(WeightTextBox1.Text)**

**ElseIf rdoPound.Checked = True Then**

**weight = Val(WeightTextBox1.Text) / 2.205**

**weightBMI = Val(WeightTextBox1.Text) \* 0.453592**

**End If**

**If cmbActivityLevel.Text = "Lightly (Little or No Exercise)" Then**

**activityLevel = 1.3735**

**ElseIf cmbActivityLevel.Text = "Moderately (2-5 times/week)" Then**

**activityLevel = 1.5488**

**ElseIf cmbActivityLevel.Text = "Very Active (6-7 times/week)" Then**

**activityLevel = 1.7234**

**ElseIf cmbActivityLevel.Text = "Extra Active (Intense Exercise Daily)" Then**

**activityLevel = 1.898**

**End If**

**lblMaintain.Text = Food.CalorieFormula(Val(AgeTextBox1.Text), GenderComboBox1.Text, height, weight, activityLevel)**

**lblLose1.Text = Val(lblMaintain.Text) - 500**

**lblLose2.Text = Val(lblMaintain.Text) - 1000**

**lblGain1.Text = Val(lblMaintain.Text) + 500**

**lblGain2.Text = Val(lblMaintain.Text) + 1000**

**BMI = Food.BMIFormula(weightBMI, heightBMI)**

**lblBMI.Text = Format(BMI, "0.00")**

**Food.BMICategory = BMI**

**lblBMICategory.Text = Food.BMICategory**

**End Sub**

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

**If txtAddedFood\_ID.Text = "" Then**

**MsgBox("Please Select an Item First!")**

**Else**

**Dim ask = MsgBox("Are you sure you want to Delete this Item?", MsgBoxStyle.YesNo)**

**If ask = MsgBoxResult.Yes Then**

**DataGridView2.Rows.Remove(DataGridView2.SelectedRows(0))**

**MyStoredProcedures.DeleteHistory(txtAddedFood\_ID.Text)**

**MsgBox("Deleted Successfully!")**

**Me.Tbl\_AddedFoodTableAdapter.Fill(Me.UserAccountDataSet.tbl\_AddedFood)**

**End If**

**End If**

**End Sub**

**Private Sub DataGridView2\_CellClick(sender As Object, e As DataGridViewCellEventArgs) Handles DataGridView2.CellClick**

**Dim selectedRowIndex As Integer**

**selectedRowIndex = e.RowIndex**

**Dim row As New DataGridViewRow**

**If selectedRowIndex > 0 Then**

**row = DataGridView2.Rows(selectedRowIndex)**

**txtAddedFood\_ID.Text = row.Cells(0).Value.ToString()**

**End If**

**End Sub**

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

**Dim ask = MsgBox("Are you Sure you Want to Delete All History?", MsgBoxStyle.YesNo)**

**If ask = MsgBoxResult.Yes Then**

**MyStoredProcedures.DeleteAllHistory()**

**Me.Tbl\_AddedFoodTableAdapter.Fill(Me.UserAccountDataSet.tbl\_AddedFood)**

**MsgBox("Deleted All History")**

**End If**

**End Sub**

**Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click**

**Account.getName = Full\_NameTextBox.Text**

**Account.getGender = GenderComboBox.Text**

**Account.getAge = Val(AgeTextBox.Text)**

**Account.getHeight = Val(HeightTextBox.Text)**

**Account.getWeight = Val(WeightTextBox.Text)**

**If (Full\_NameTextBox.Text = "" AndAlso AgeTextBox.Text = "" AndAlso HeightTextBox.Text = "" AndAlso User\_NameTextBox.Text = "" AndAlso txtConfirmPass.Text AndAlso User\_PasswordTextBox.Text = "") = True Then**

**MsgBox("Please Complete All Missing Boxes.", MsgBoxStyle.OkOnly, "Unsuccessful!")**

**ElseIf Account.getAge = -1 Then**

**MsgBox("You're Too Young to Use This Application!", MsgBoxStyle.OkOnly, "Invalid!")**

**ElseIf Account.getAge = 0 Then**

**MsgBox("You're Too Old to Use This Application!", MsgBoxStyle.OkOnly, "Invalid!")**

**ElseIf Account.getHeight = 0 Then**

**MsgBox("Height Input is Either Too Low or Too High." & vbNewLine & "Please Check Again!", MsgBoxStyle.OkOnly, "Invalid!")**

**ElseIf Account.getWeight = 0 Then**

**MsgBox("Weight is too low.", MsgBoxStyle.OkOnly, "Invalid!")**

**Else**

**If User\_PasswordTextBox.Text Like txtConfirmPass.Text = True Then**

**Dim ask = MsgBox("Are You Sure You Want to Update Account Details?", MsgBoxStyle.YesNo, "[CalorieIntakeManager]")**

**If ask = MsgBoxResult.Yes Then**

**MyStoredProcedures.UpdateAccount(Val(User\_IDTextBox.Text), Account.getName, Account.getGender, Account.getAge, Account.getHeight, Account.getWeight, User\_NameTextBox.Text, User\_PasswordTextBox.Text)**

**MsgBox("Your Account Has Been Updated Successfully.", MsgBoxStyle.OkOnly, "Success!")**

**End If**

**Else**

**MsgBox("User Password and Confirm Password are not the same." & vbNewLine & "Please Try Again.", MsgBoxStyle.OkOnly, "Unsuccessful!")**

**End If**

**End If**

**End Sub**

**Private Sub rdoM\_CheckedChanged(sender As Object, e As EventArgs) Handles rdoM.CheckedChanged**

**If rdoM.Checked Then**

**Label21.Text = "m"**

**Account.getHeight = Val(HeightTextBox.Text)**

**HeightTextBox.Text = Account.Divide(Account.getHeight, 3.281)**

**ElseIf rdoFT.Checked = True Then**

**Label21.Text = "ft"**

**Account.getHeight = Val(HeightTextBox.Text)**

**HeightTextBox.Text = Account.Multiply(Account.getHeight, 3.281)**

**End If**

**End Sub**

**Private Sub rdoLB\_CheckedChanged(sender As Object, e As EventArgs) Handles rdoLB.CheckedChanged**

**If rdoLB.Checked Then**

**Label22.Text = "lb"**

**Account.getWeight = Val(WeightTextBox.Text)**

**WeightTextBox.Text = Account.Multiply(Account.getWeight, 2.205)**

**ElseIf rdoKG.Checked = True Then**

**Label22.Text = "kg"**

**Account.getWeight = Val(WeightTextBox.Text)**

**WeightTextBox.Text = Account.Divide(Account.getWeight, 2.205)**

**End If**

**End Sub**

**Private Sub CheckBox1\_CheckedChanged(sender As Object, e As EventArgs) Handles chShowPass.CheckedChanged**

**If chShowPass.Checked Then**

**User\_PasswordTextBox.UseSystemPasswordChar = False**

**txtConfirmPass.UseSystemPasswordChar = False**

**Else**

**User\_PasswordTextBox.UseSystemPasswordChar = True**

**txtConfirmPass.UseSystemPasswordChar = True**

**End If**

**End Sub**

**End Class**

**AccountClass.vb**

Public Class AccountClass

Private FullName, Gender As String

Private Age As Integer

Private height, weight As Double

Public Property getName As String

Get

Return FullName

End Get

Set(value As String)

FullName = value.ToUpper()

End Set

End Property

Public Property getGender As String

Get

Return Gender

End Get

Set(value As String)

If value = "FEMALE" Then

Gender = "FEMALE"

ElseIf value = "MALE" Then

Gender = "MALE"

End If

End Set

End Property

Public Property getAge As Integer

Get

Return Age

End Get

Set(value As Integer)

If value < 10 Then

Age = -1

ElseIf value > 99 Then

Age = 0

Else

Age = value

End If

End Set

End Property

Public Overridable Property getHeight As Double

Get

Return height

End Get

Set(value As Double)

If value < 1 AndAlso value > 9 Then

height = 0

Else

height = Format(value, "0.00")

End If

End Set

End Property

Public Overridable Property getWeight As Double

Get

Return weight

End Get

Set(value As Double)

If value < 30 Then

weight = 0

ElseIf value = Nothing Then

weight = 0

Else

weight = Format(value, "0.00")

End If

End Set

End Property

Public Function Multiply(input As Double, value As Double) As Double

input \*= value

Return Format(input, "0.00")

End Function

Public Function Divide(input As Double, value As Double) As Double

input /= value

Return Format(input, "0.00")

End Function

End Class

**FoodClass.vb**

Public Class FoodClass

Inherits AccountClass

Implements CalculateCaloriesInterface

Private food\_Search As Integer

Dim categoryBMI As String

Public Property foodSearchID As Integer

Get

Return food\_Search

End Get

Set(value As Integer)

If value >= 0 AndAlso value <= 1000 Then

food\_Search = value

ElseIf value = Nothing Then

food\_Search = 1

Else

food\_Search = 0

End If

End Set

End Property

Public Overrides Property getHeight As Double

Get

Return MyBase.getHeight

End Get

Set(value As Double)

If value < 3 Then

MyBase.getHeight = 0

ElseIf value > 3 Then

MyBase.getHeight = -1

End If

End Set

End Property

Public Overrides Property getWeight As Double

Get

Return MyBase.getWeight

End Get

Set(value As Double)

If value < 100 Then

MyBase.getWeight = 0

ElseIf value > 100 Then

MyBase.getWeight = -1

End If

End Set

End Property

Function CalorieFormula(Age As Integer, Gender As String, Height As Double, Weight As Double, ActivityLevel As Double) As Integer Implements CalculateCaloriesInterface.CalorieFormula

Dim total As Integer

If Gender = "MALE" Then

total = ((10 \* Weight) + (6.25 \* Height) - (5 \* Age) + 5) \* ActivityLevel

ElseIf Gender = "FEMALE" Then

total = ((10 \* Weight) + (6.25 \* Height) - (5 \* Age) - 161) \* ActivityLevel

End If

Return total

End Function

Function BMIFormula(Weight As Double, Height As Double) As Double Implements CalculateCaloriesInterface.BMIFormula

Dim BMIOUtput As Double

Height = Height \* Height

BMIOUtput = Weight / Height

Return BMIOUtput

End Function

Public Property BMICategory As String

Get

Return categoryBMI

End Get

Set(ByVal value As String)

If (value < 18.5) Then

categoryBMI = "Underweight"

ElseIf (value >= 18.5 And value <= 24.9) Then

categoryBMI = "Normal Weight"

ElseIf (value >= 25 And value <= 29.9) Then

categoryBMI = "Overweight"

ElseIf (value >= 30) Then

categoryBMI = "Obese"

End If

End Set

End Property

End Class

**CalculateCaloriesInterface.vb**

Public Interface CalculateCaloriesInterface

Function CalorieFormula(Age As Integer, Gender As String, Height As Double, Weight As Double, ActivityLevel As Double) As Integer

Function BMIFormula(Weight As Double, Height As Double) As Double

End Interface