# Introduction

ESC GymApp is a Windows Forms application written in C# using Visual Studio 2017. We were tasked to produce a piece of software using the contextual scenario below and follow all professional computing standards and good practice procedures. The finished solution consists of the frontend (GUI) and the backend (code).

# Scenario

You are asked to use your software design, development, testing and evaluation understanding and skills to produce a program that meets the client’s requirements.

East Surrey College gym has commissioned you as a software developer to write a program that will assess a gym member’s requirement to maintain their current weight accurately.

You need to create a program that will give the learners gym member information about:

• Their current basal metabolic rate (BMR)

• Their current body mass index (BMI)

• Their target BMI

• The number of kilocalories to maintain their current weight.

# Scope

## In Scope

## The program calculates and displays a gym member's basal metabolic rate (BMR)

## Calculates and display the gym member's body mass index (BMI)

## The number of kilocalories to maintain their current weight.

## The BMR calculation is given to 2 decimal places.

## The BMI calculation is given to 1 decimal place.

## The kilocalorie requirement output is shown rounded to a whole number.

## The program needs a "login" screen to access the application.

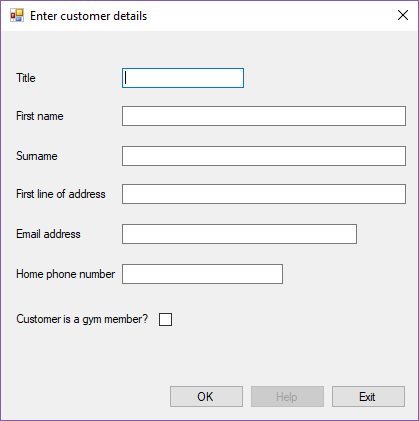
## A screen to collect customer details, if they're an existing gym member, name, address, email and telephone.

## Non-Functional Requirements

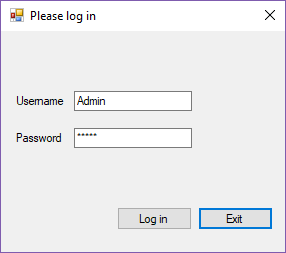
## *Must be Windows Form application written in C# & Java*

# Form design

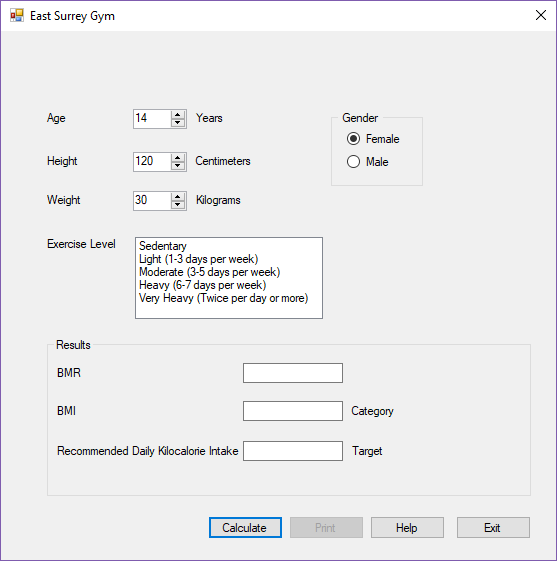
## Customer Details Form



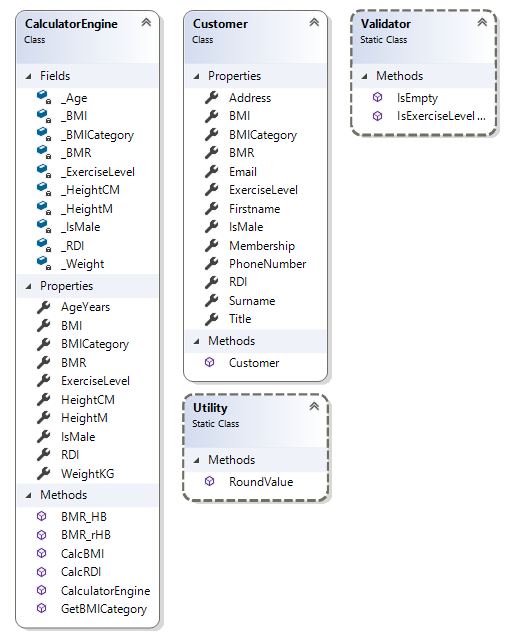
## Login Form



## Gym Form

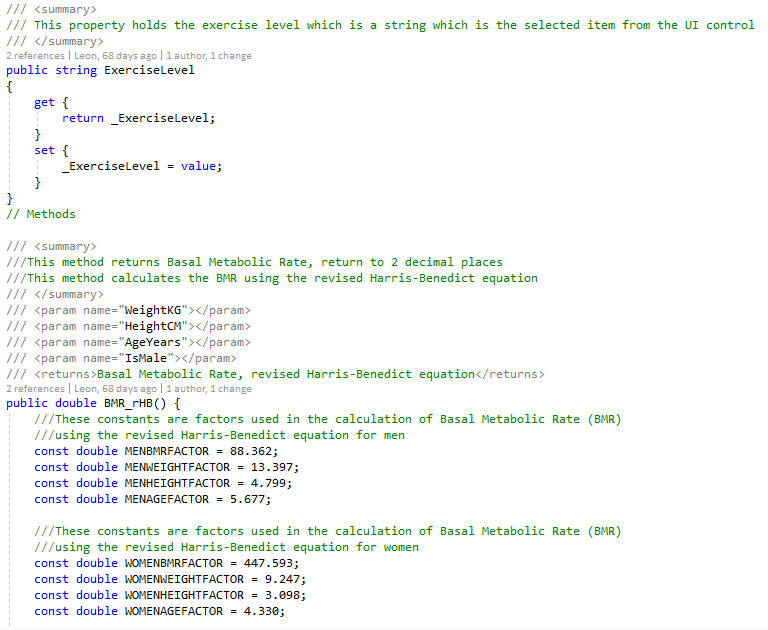


# Class Diagram

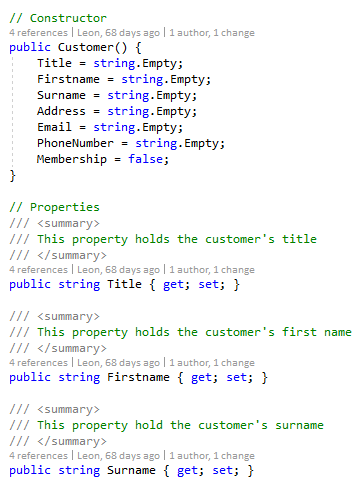


# Classes

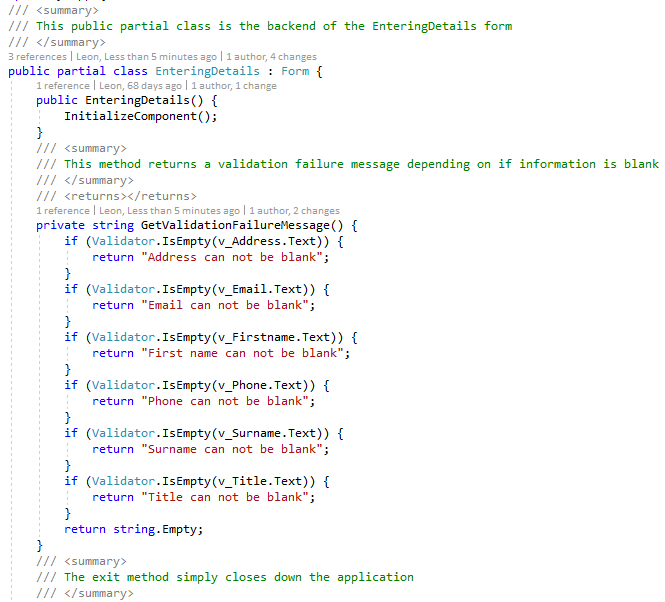
## CalculatorEngine.cs



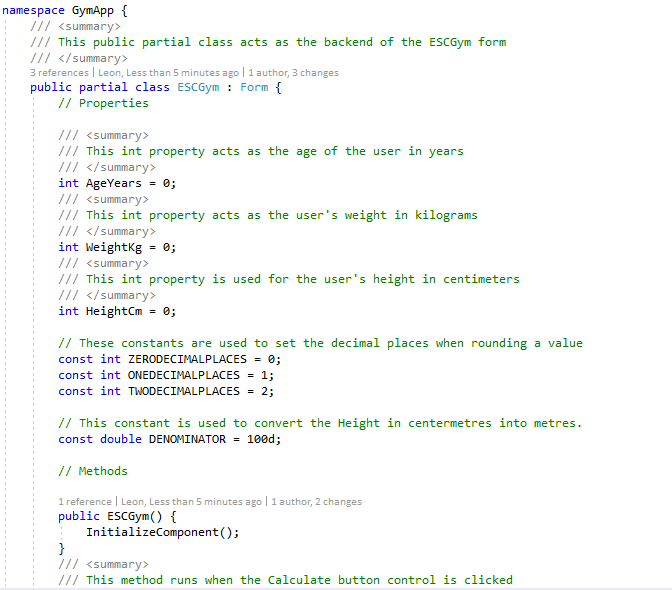
## Customer.cs



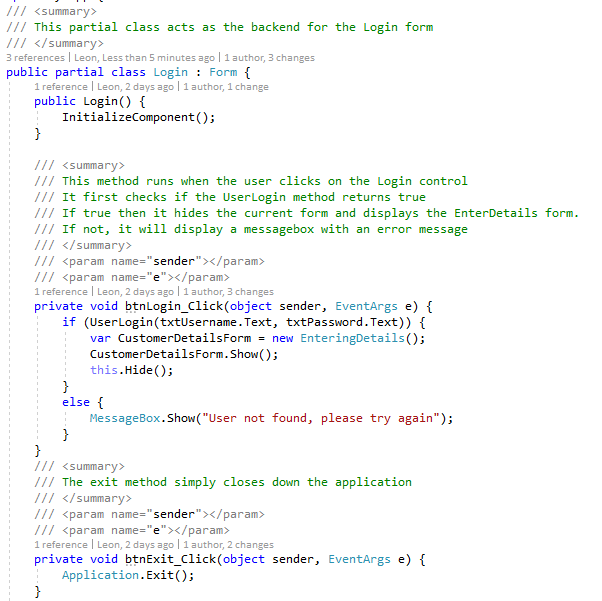
## EnteringDetails.cs



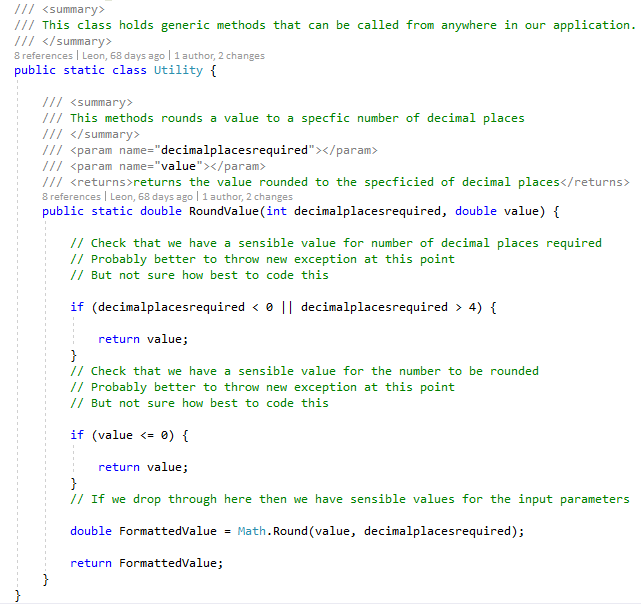
## ESCGym.cs



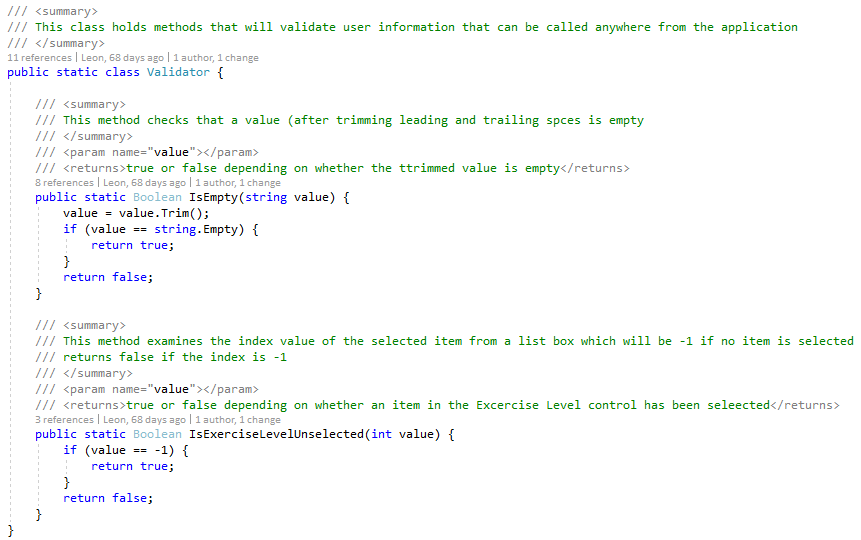
## Login.cs



## Utility.cs



## Validator.cs



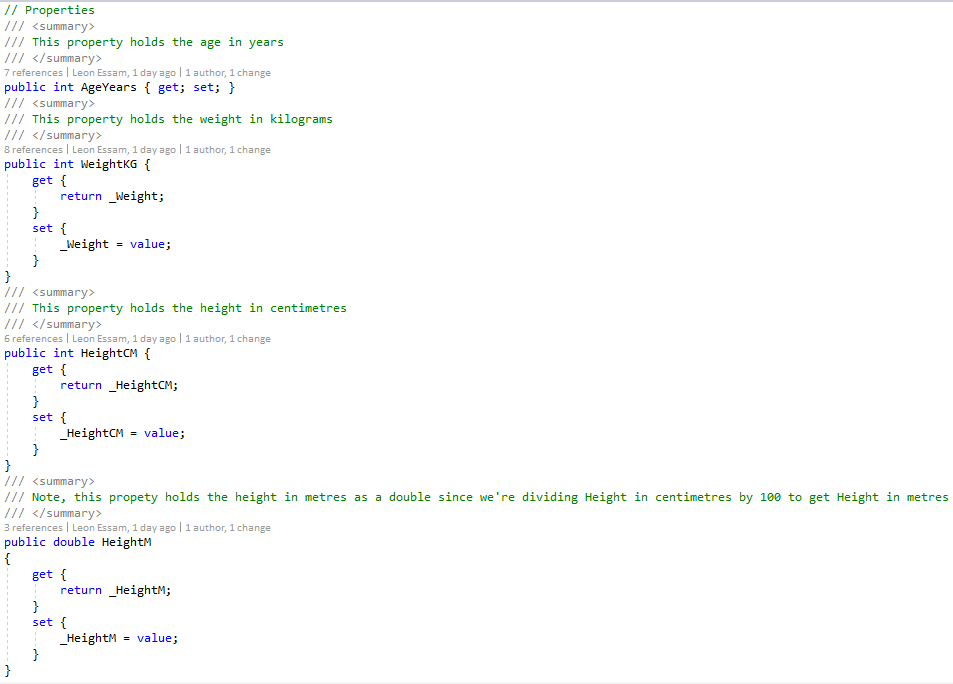
# Properties

## Adding properties

The first initial application didn't have any properties, instead it relied on public static variables in different classes to store and access the information. The revised application however now uses properties and fields in classes that must be instantiated. This is good because now multiple customers can be instantiated with different information instead. It is also useful for alter development like printing out a list of customers.

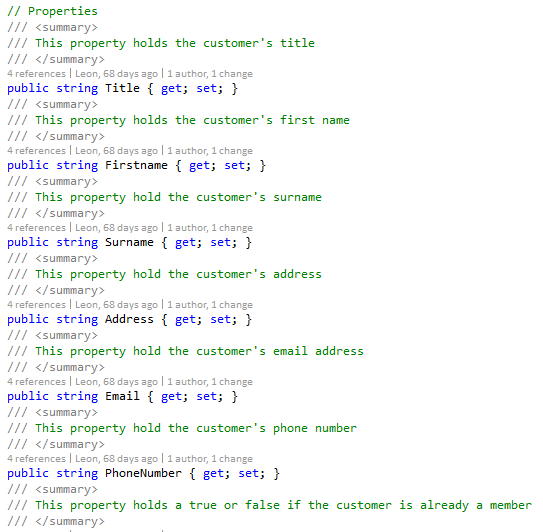
## CalculatorEngine properties

These properties are used in calculations of BMI, BMR and other information. Customer information is fed to this class which in turn is used in multiple methods and then assigned to new properties like BMI and BMR, which are used in different classes.



## Customer properties

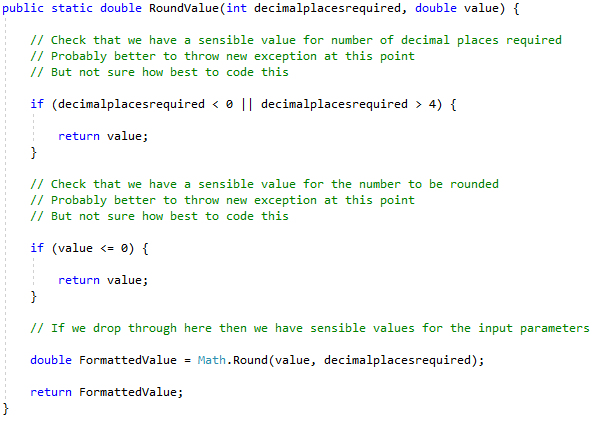
Customer properties include personal information like name, address and email. These are used to store BMI and other information in the main form.



# Methods

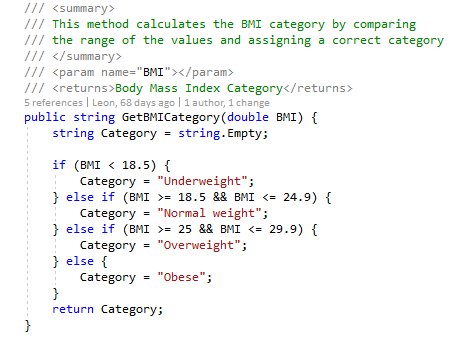
## RoundValue : double

This is a simple method in the Utility class, it takes a value and how many decimal places the value should be rounded to. I’ve done it this way as it makes maintaining easier and avoids hardcoding values.



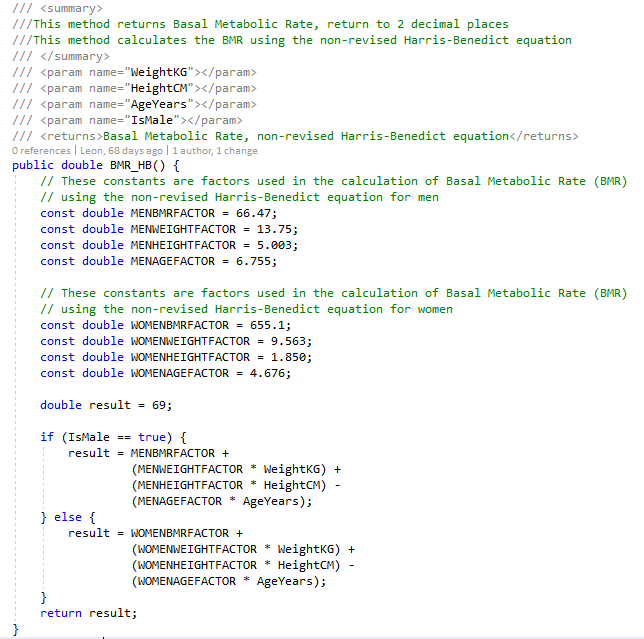
## GetBMICategory : string

This method checks the value of the BMI and returns a string accordingly for their current weight category. This method is easy to maintain since we know the BMI requirements from the scenario, we can use hardcoded values to return the correct BMI category.



## BMR\_HB : double

This method is similar for calculating the BMR, this method however uses the non-revised Harris-Benedict equation, so this allows the gym staff to now choose one of two equations.



# Testing

## NUnit - Unit Tests

## CalculatorEngineTest.cs

## CustomerTest.cs

## UtilityTest.cs

## ValidatorTest.cs

# Review

## Richard’s Review

## What went well

### Development

### Database Schema

### QA Testing

## What didn't go well

### Project Management

### Development

### QA Testing