Introduction

Design

Table of Contents

**Introduction1**

**Introduction**

**.**2

Type chapter title (level 3)3

**Type chapter title (level 1)4**

Type chapter title (level 2)5

Type chapter title (level 3)6

Flowchart

UML

Classes

Class:menu

Class:patient

Data types

Implementation

Testing

Test 1adding a person

Test 2 updating a persons weight

Test 3 update contact

Test 4 display patient information

Test 5 display patient past BMI

Test 6 sort patients

Test 7load from file

Test 8 save to file

# Introduction

The client brief which has been chosen was the BMI calculator. This decision was because the overall design was the only one I had a good idea of completing upon reading, as my plan had changed very little throughout the coding process and the creation of this report.

Aspects that shall be covered within this report will include the plan and design process, the implementation of the code, screenshots of the code doing what was asked on the brief, and extra features which I had included.

# Design

## Flowchart

A screenshot of a computer

Description automatically generated with medium confidence

This flowchart indicates the given paths of the code, all the aspects that had been decided to include to be implemented, and the plan is designed to show that a menu loop will be created.

## UML

This UML plan shows all the functions currently included within the code.

The UML had to be updated and edited a lot as many get and set functions had to be created for the load and write to aspects which were asked for.

A screenshot of a computer

Description automatically generated

## Data types

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data type | Example Data | Reasoning |
| Keep\_menu\_going | Boolean | True | Because for the menu, the loop either continues or it stops and quits |
| Height | Float | 5.5 | Height isn’t a whole number |
| Weight | Float | 4.6 | Weight isn’t a whole number |
| bmidata | float | 21.56 | Created by weight and height, so it’s a float. And BMI isn’t always a whole number. |
| weights | Vector<float> | {5.5,12.6,78.4} | It contains a collection of the weights variable, so it must be the same data type. |
| BMIs | Vector<float> | {2.6,8.9,6.36} | It is a collection of BMI, so it must be the same data type as BMIdata. |
| Patients | Vector<Patients> | {height:1.66,  weights<float>:{6.51,6.5,7.4}  email:”my name is bob”  } | Because this vector contains multiple instances of the class patient and its respective information |
| Name | String | Mason | Names are string |
| Phone number | string | 0292070217 | This is only being stored, and no calculations are being made; also, it could start with +44 |
| Email | String | [James.bond@hotmail.com](mailto:James.bond@hotmail.com) | The email contains special characters |

# Implementation

A screenshot of a computer

Description automatically generated

This is the patient class. The variables and functions have been implemented just as stated in the design. The weights are stored within a vector so that an indefinite number of weights can be stored in this class.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

These are images of the main class, which includes all the movement between functions from the patient class and any position changes made within any vectors.

This code is most definitively the most enjoyable to witness as it sorts the patients by BMI in order of their most recent weight inputs. When saved to the file, it’ll be in the order of the sorted list, so the only time the sort will need to be done again is when another weight is added, or another patient has been created.

# Testing

## Test 1 Adding a person

Step 1: Select create a new patient

Step 2: Input requested information

Step 3: View the list of patients

Evidence of test pass:

Adding information

A screenshot of a computer

Description automatically generated

Viewing data

A screenshot of a computer

Description automatically generated with medium confidence

This only displays the patient information and not the BMI, weight or height since that is classed as patient medical information.

## Test 2 updating a person weight

Step 1: Select update patient weight from the menu:

Step 2: enter updated patient weight

Step 3: View the patient to see the new and old weight.

Evidence of test pass:

Single weight input and one BMI output

Graphical user interface

Description automatically generated

Updated weight input and additional BMI output:

A screenshot of a computer

Description automatically generated with medium confidence

## Test 3 update contact information

Step 1:select update contact information

Step 2: choose the patient whose information is to be updated

Step 3: display patient new information

Evidence of success:

New information inputted:

Graphical user interface

Description automatically generated

New information displayed:

Graphical user interface

Description automatically generated

## Test 4 displays patient information

Step 1: select display patient information

Step 2: display patient information

Evidence of success:

Graphical user interface

Description automatically generated

## Test 5 displays the patient past BMI

Step 1:select display patient history BMI

Step 2:select patient to display

Graphical user interface

Description automatically generated

Evidence of history BMI:

A screenshot of a computer

Description automatically generated with medium confidence

## Test 6 sort patients

Step 1:select sort patients

Step 2:display patient results

An aspect that must be considered is that there must be two or more patients already in the system for sorting.

Creating new patients to be sorted for this part:

A screenshot of a computer

Description automatically generated

Graphical user interface

Description automatically generated

Evidence of the sort working:

The first display shows all the patients in the order they were inputted.

Once the number 6 is entered, the second indicates that Francis is larger than George.

A screenshot of a computer

Description automatically generated with medium confidence

## Test 7 Load from file

Step 1: display patients ( will show no current patients)

Step 2: select 7 to load in from the file

Step 3 display the patients that exist

Evidence of the load:

Clean patient list

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

## Test 8 save to file

Unfortunately for the 8th test, which I would show by here, it has failed as of right now. I have an issue with a loop of mine that keeps incrementing with no regard for any other variables and is not being called anywhere else by any other function. And because of this, if I were to save the current information into the document, the wrong inputs would be saved.

# Conclusion

In conclusion, the contents of this report have considerably gone align with the criteria of the brief selected and achieved most aspects asked to be completed. An area within brief that is believed to be improved upon would be the save to file element requested, which had failed miserably. Although that may be the case, the focus of the brief has been achieved, and the knowledge and understanding which this report has been significantly provided.

The save to file function had issues:

A screenshot of a computer

Description automatically generated

And the function which I believe to be the leading cause of the problem:

A screenshot of a computer

Description automatically generated