# Calories and nutrients calculator

## 1. Personal information

Rami Lahtinen, 916929, sähkötekniikan koulutusohjelma, 3rd year, 22.2.2024

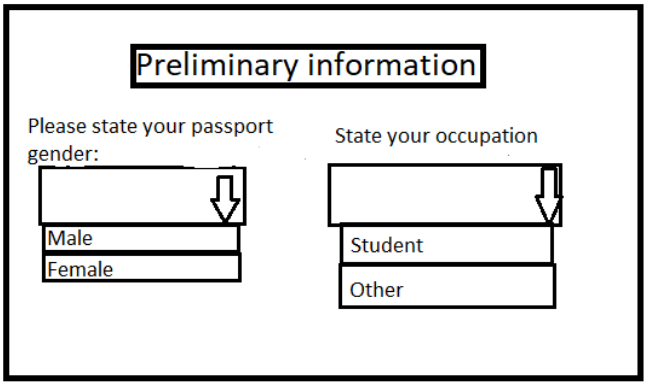
## 2. Overall description

This program allows the user to monitor their nutrition and energy intake. It holds different nutrition goals for different archetypes of people, such as an office worker or a student. It will also be able to monitor a few micronutrients in addition to the macro level. My goal for the project is to create a program that is robust and as to use as possible so that anyone understands what is happening without prior reading. I am envisioning completing the project on the “medium”-level so that my program would have a working graphic UI. The program would also save an individual's goals to a text file.

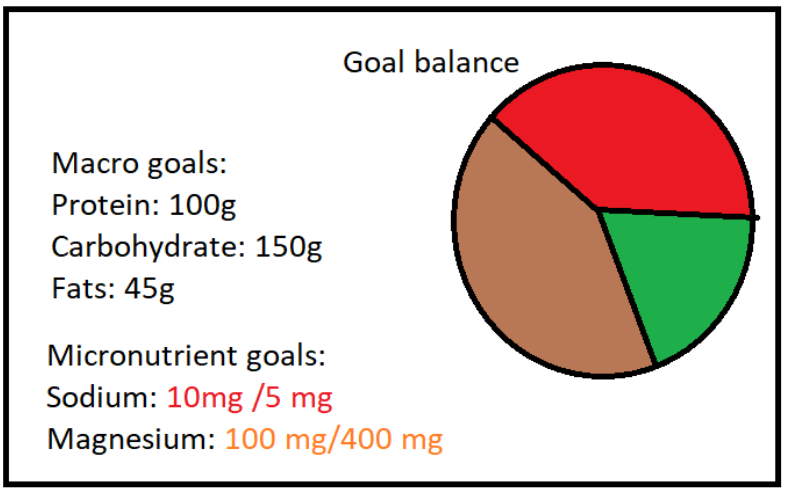
## 3. Use case and UI

At first the program will probably only have a text-based UI as outlined in the Easy stage of the project. Later a graphic UI will be implemented and all the details will be conveyed either via buttons, sliders, drop-down menus or text inputs. The user will be presented with a goal of energy and nutrients, and they can input foods that contribute towards these goals with some sort of progress bar or a numeric representation. When the user goes over a goal, the text turns red.

When the program starts the user will be asked preliminary information, presented below:



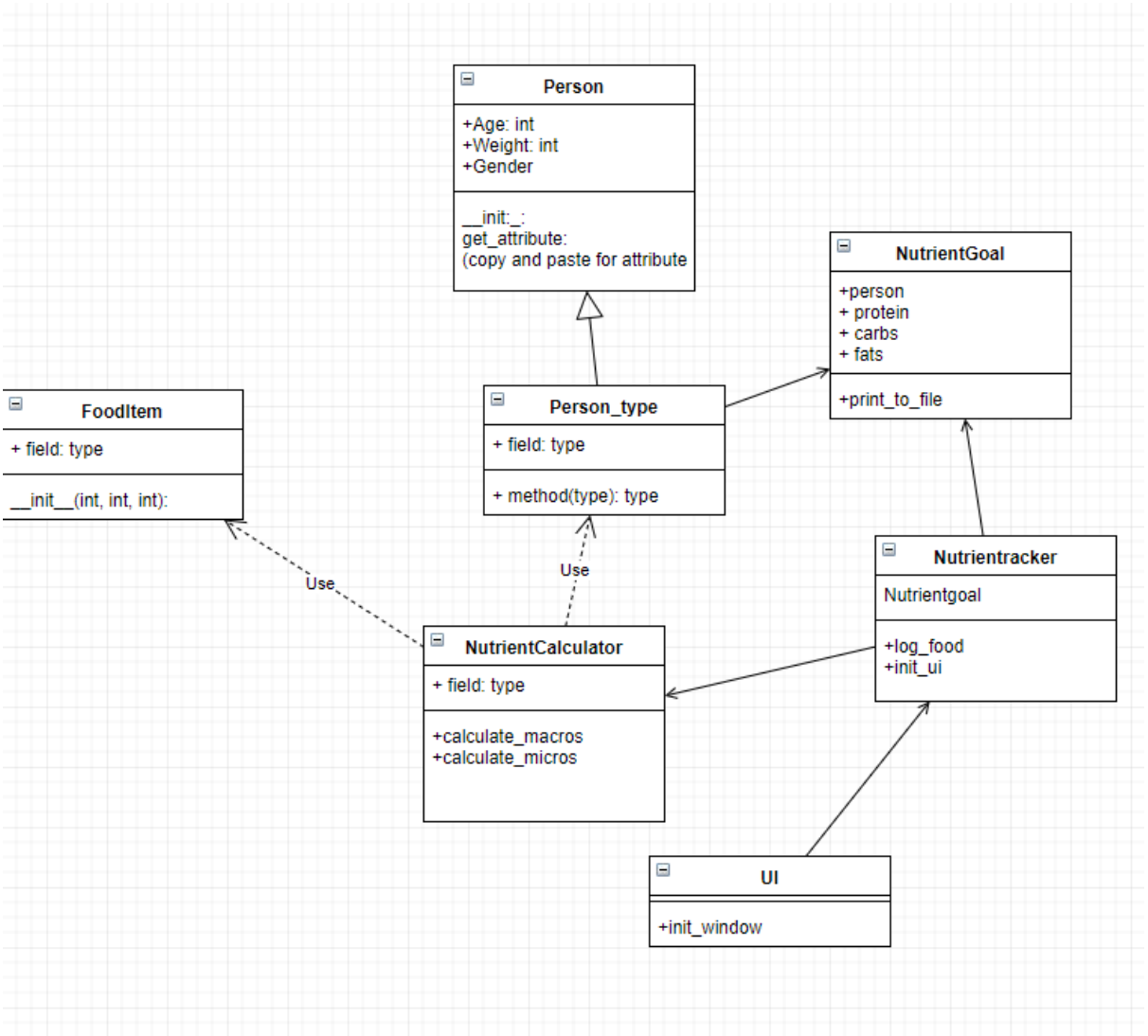

This information will also include a daily activity estimate and so on.
When all the information has been gathered, a window with goals will show:




I am not yet sure how to present the different macronutrient goals based on the user input, but I am sure it will come with time.

## 4. UML Plan

Please note: this is, by no means, a final class structure for the program but a possible version of what the class structure would look like. First of all, we need a Person class and subclasses for the different archetypes of people. We also need a class for the UI, a NutrientGoal class, a FoodItem class and maybe a calculator class. Finally, we probably need a tracker class that is in charge of forwarding all the info into the UI class.




I do not know all of the methods that will be needed in the final program, this is just a representation and is subject to changes further along the development path.

## 5. Data Structures

For this project I am most likely going to need dynamic data structures[1], especially if the program keeps track of its users. That would warrant the use of a list[2] to store the different users into. Many of the variables are probably going to be stored as a static integer or a dictionary. These include the values for the different nutrients, both macro and micro.

## 6. Files and formats

The data that the program saves is at least partly going to be written in a text file from where a user can have an overview of their goals and the inputted foods. Data in the file is going to be structured as “Datapoint: value” until the whole range of different parameters has been printed into the text file. The program could also read these types of files and forward the information straight to the UI without the need to input all of the data by hand.

# 7. Algorithms

Energy needs are calculated based on the person archetype a user provides and calculated using the formulas developed by the Institute of Medicine[3]. The formulas for energy needs are as follows:
Males:
$E = 662 - (9,53 * A) + PA * ((15,91 * W) + (539,6 * H))$
Females:
$E = 354 - (6,91 * A) + PA * ((9,36 * W) + (726 * H))$

In the formulas A stands for age in years, W for weight in kilograms and H for height. PA stands for physical activity and is a fixed value:

| **PA** | **Males** | **Females** |
|---|---|---|
| Sedentary | 1,0 | 1,0 |
| Low Active | 1,11 | 1,12 |
| Active | 1,25 | 1,27 |
| Very Active | 1,48 | 1,45 |

These are crucial for the program's successful running, since these are the bread and butter of the program. Almost all of the information that a user inputs into the program will be running through these calculations at some point.

The energy of macronutrients is defined as follows: Protein: 4kcal/g, Carbohydrate: 4kcal/g, Fats: 9kcal/g (Institute of Medicine). This information forms the other side and the balance calculations in the program.

# 8. Testing plan

In the testing phase it is obvious to make sure that the error handling works in every part of the program. The first one that comes to mind is the correct writing and reading of files, since they form a very important aspect of the project. Also, correct inputs are to be tested so that a person can't input their age as "A" or anything along the lines of that. The same holds true for all of the preliminary information a user inputs into the program. It is also paramount to test all the methods that contribute to the energy and nutrient goals of the user. I will be utilising unit tests[4], most likely AssertEqual to determine whether the program works or not.

# 9. Libraries

I will be using PyQt6[5] as the instructions permit. I am, however, very unsure if the sublibraries of PyQt6 are permitted or not. I will be using the permitted sublibraries, most likely QtWidgets[6] and parts of it such as QMainwindow or QVBoxlayout.

## 10. Timetable

This is a very rough estimate of the hours that are required to complete the project and all of the various parts in the process. First is, of course, planning. This project plan is not a complete one at least at this time due to strict time constraints. Refining the plan will probably take me somewhere along the lines of 10-12 hours.

For the first checkpoint (3) there will be at least a very basic working version of the code with a text-based interface for one type of person.

For the second checkpoint (6) there will be multiple person archetypes and at least two unit tests for different parts of the program.

The actual coding of the program will probably take me a long time. A figure that comes to mind is around 80-100 hours. I don't regard myself as a very skilled programmer yet, so I fear that little mistakes and bugs will take me frustratingly long to fix. The lessons of the course Y2[7] have heavily contributed how I plan on completing the project.

## 11. References and links


1. Ramalho, Luciano. *Fluent Python*. Sebastopol, CA: O'Reilly, 2015. Print.
2. Python Data Structures https://docs.python.org/3/tutorial/datastructures.html
*3. Dietary Reference Intakes for Energy, Carbohydrate, Fiber, Fat, Fatty Acids, Cholesterol, Protein, and Amino Acids*. Washington, DC: National Academies Press, 2005. Print.
4. Unit tests https://docs.python.org/3/library/unittest.html
5. PyQt6 https://www.riverbankcomputing.com/software/pyqt/
6. QtWidgets https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/index.html
7. Course home page https://plus.cs.aalto.fi/y2/2024/