Python Project Documentation

Barry O'Connor [N0813926]

COMP10082 Programming Strand Assignment

1 Aims

This project is an attempt to create a nutrition tracker application in Python. There are several, feature-rich versions of these already available on phones but they are beyond the scope of this project. This project will concentrate on creating an application with a basic but functional feature list, which can fulfil the core needs of anyone wishing to either keep a food diary or track their food intake.

Macronutrients (“Macros”) refer to the Protein, Carbohydrate and Fat content of a food. Combined with calories, these four values are used by anyone wishing to diet, gain muscle or maintain their weight. At a basic level the application must allow the creation of a record of food intake on a daily basis and allow the user to browse this information.

Since Macros vary depending on personal needs and are likely to change as a user progresses, there must be an editable record of these goals so that the user can amend these based upon their current needs.

These actions will create a potentially large amount of information so some form of permanent storage will need to be used so that the information is saved and can be used on an ongoing basis.

Since the food diary and goals will be available, it would make sense that there be some form of overview which would allow the user to quickly see how they are progressing towards their goals for the day.

2 Requirements Analysis

2.1 User Interface Requirements

The application will require the user to input a significant amount of data. Even at the most basic level they will be required to input values for all 4 macro values every time they wish to add an item of food to the application. This could potentially be achieved using the Python command line interface but doing so might become cumbersome and frustrating for the user. Considering the already somewhat tiresome routine of counting values for everything eaten, this may prove to be an excuse for the user to stop using the application

A graphical interface would allow for the creation of forms and inputs such as text boxes, comboboxes and buttons. This format would allow a user to enter values with a mouse and keyboard in a more visual style. This will be very familiar to users making the experience a more comfortable one. This would also allow for the display of a lot of information in a relatively small space and potentially the use of graphical representations of goals.

Python 3 includes with a Graphical User Interface library ("tkinter") which will be used to create the interface.

2.1 Storage Requirements

The application will need to store increasing lists of food entries since the user will be inputting more each day. This information not only needs to be stored securely, it also needs to be easily searched and used for comparison against the user's goals. The storage aspect could be carried out in many ways: text files, data files or a database would all suffice for storage. In terms of accessing and comparing the data quickly and efficiently, it makes sense to use a database which would allow for the querying of large amounts of records.

Python 3 includes the SQLite3 library which allows for the usage of file-based SQLite databases, this will be used to store the data for the application.

2.1 Functional Requirements

The broad aims of the application can be broken down into smaller, more defined requirements which will shape the functionality of the application. These basic requirements must be met for the application to work as a nutrition tracker. Additional features are suggested should there be enough time after completing the minimum requirements.

The Minimum Requirements for the application are:

- Account Based:
  - An account for each user. This allows for privacy and the personalisation of goals on an individual basis.
  - A login screen so that the user can access their account.
  - A registration form to allow for the creation of new accounts.
  - An account page where a user can update their personal goals at any time, this will then reflect throughout the application.

- Food Diary
    - The food diary must present the entries on a daily basis. The user must be able to change the date to allow for the navigation of the diary.
    - The entries on the diary should present an overview of the day which is easily readable.
    - The user must be able to create food entries which require the following values: food name, weight of the food eaten, calories, protein and fat values of the food.
    - Since mistakes can happen, food entries must be easily editable and deletable.

- Goal overview
    - There should be a summary of the day's progress towards each goal. Ideally this would be the first screen after the login process so the user can immediately see their daily progress.

Suggested additional features:

- Account Based:
    - Create a way of using the user's weight and height to work out their macros programmatically using equations such as Mifflin St Jeor or Klatch McArdle.

- Food Diary
    - Create a way to import and export the diary for backup purposes

- Goal overview
    - Colour code the overview so that the goals change colour depending on whether the user has reached the goal or gone over their requirements.
    - Investigate ways of adding a stacked bar chart to display goals in a clear, graphical format.

3 Design

3.1 Database Diagram

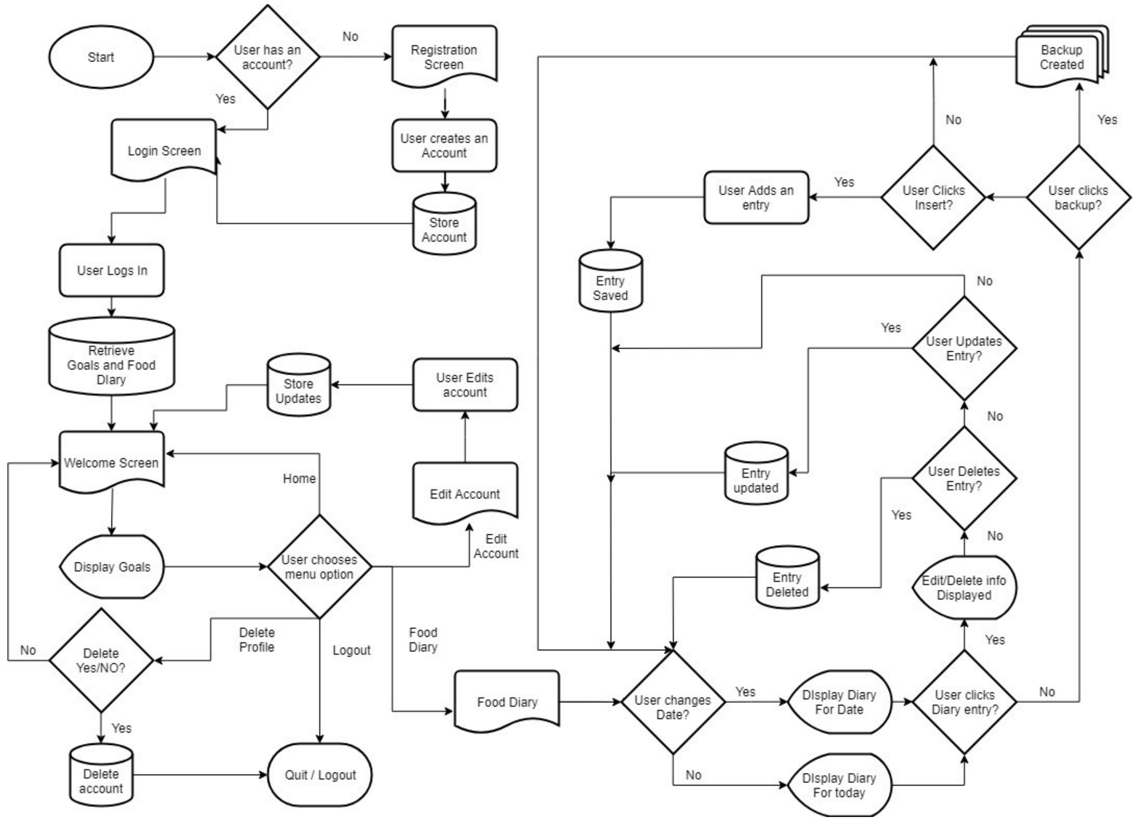
The database consists of the following two tables, one for Users and one for the Food Diary. They have a one to many relationship, with theusers.id field being used as a foreign key in the food_diary.user_id field.

food_diary Table				
Fieldname	**Type**	**Description**	**Null**	**Info**
id	integer	Unique record ID	NO	PK Unique Auto Increment
date	text	Date in format “dd-mm-yyyy”	NO	
user_id	text	user_id (from users table)	NO	
food_name	real	Name of the food eg “Bacon”	NO	
food_weight	real	Weight of the food in grams	NO	
calories	real	Number of calories in the serving	NO	
protein	real	Protein in the serving (grams)	NO	
carbs	real	Carbs in the serving (grams)	NO	
fat	real	Fat in the serving (grams)	NO	

users Table				
Fieldname	**Type**	**Type**	**Null**	**Info**
id	integer	Unique record ID	NO	PK Unique Auto Increment
username	text	username	NO	
password	text	User password	NO	
birthday	real	Date in format “dd-mm-yyyy”	NO	
weight	real	Weight of the user in kilograms	NO	
height	real	Height in centimetres	NO	
calories	real	Calorie Goal	NO	
protein	real	Daily protein goal (grams)	NO	
carbs	real	Daily carb goal (grams)	NO	
fat	real	Daily fat goal (grams)	NO	

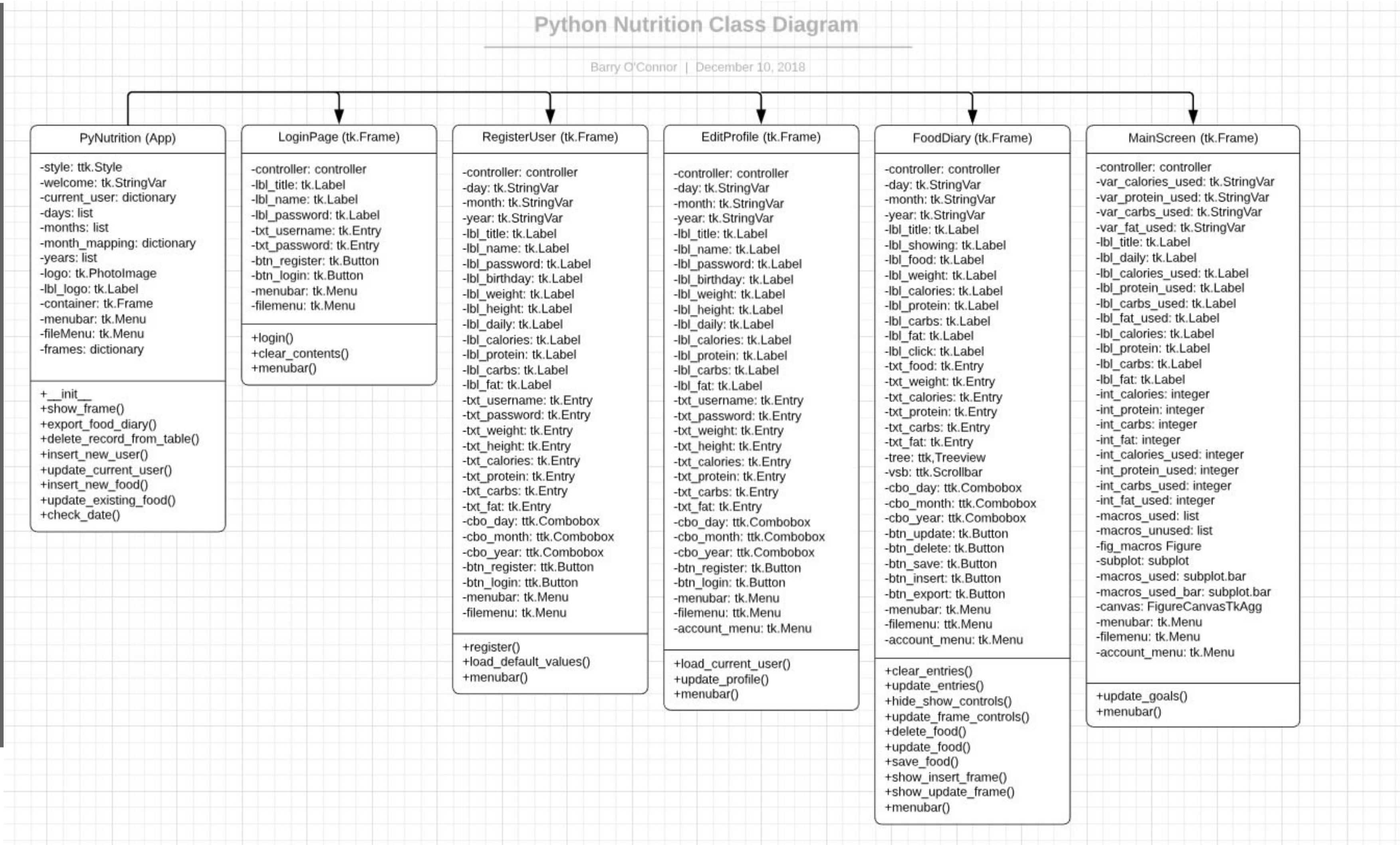
3.2 Program Flow Chart

A program flow chart for the application follows.



3.3 Classes

This application makes heavy use of the tkinter and sqlite3 libraries and the classes they include. The application uses frames which are classes and each frame contains methods and both properties and tkinter controls, which are stored as variables so can also be considered properties. The class diagrams are therefore lengthy. The classes themselves operate on a single use basis, each frame exists only once and is simply switched in and out of focus.



Python Nutrition Class Diagram

Barry O'Connor | December 10, 2018

4 Testing

During production of the application any expected issues were recorded and any unexpected bugs or issues were included as part of the review process. Where possible these have all been addressed before completion date. The following are the test cases used in testing, which may be followed independently to verify.

4.1 Test Cases

Login Page				
Test Scenario	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**
Log in with valid data	1)Open program 2)enter username 3) enter password 4) click login button	Username: barryoconnor password: pa55w0rd	Log into application successfully	Login Successful
Log in with invalid data	1)Open program 2)enter username 3) enter password 4) click login button	Username: barryoconnor password: password	Login failure with message	Login failure with message
Open Register Screen	1)Open program 4) click create account button	N/A	Redirect to Create an Account	Redirect to Create an Account

Create an Account / Edit Account Pages				
Test Scenario	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**
Create an account with defaults	1) enter username 2) enter password 3) enter weight 4) enter height 5)Click create account button	Username: usertest password: secret Weight: 76 Height:177	New user created, message shows	New user created, message shows
Create an account with random data	1) enter username 2) enter password 3) enter birthday 4) enter weight 5) enter height 6) enter calories 7) enter protein 8) enter carbs 9) enter fat 10)Click create account button	Username: usertest2 password: shhsecret Weight: 104 Height: 123 Birthday: 13/04/1984 Calories: 1764 Protein: 185 Carbs: 203 Fat: 34	New user created, message shows	New user created, message shows
Create an account with blank data	1) click create account button	N/A	Warning issued	Warning issued
Choose an invalid date	1) set date 2) click create account button	Date: 31 February 2018	Warning issued	Warning issued

Welcome/Home Page				
Test Scenario	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**
Log out	1) Click Account Menu 2) Click Log Out	N/A	Redirected to Login	Redirected to Login
Delete account but cancel	1) Click Account Menu 2) Click Delete Account 3) click No	N/A	Nothing (account remains)	Nothing (account remains)
Delete account	1) Click Account Menu 2) Click Delete Account 3) click Yes	N/A	Redirected to login and account deleted	Redirected to login and account deleted
Click Edit Account	1) Click Account Menu 2) Click Edit Account	N/A	Redirected to Edit Account	Redirected to Edit Account
Click Food Diary	1) Click Food Diary Menu	N/A	Redirected to Food Menu	Redirected to Food Menu
Click Exit	1) Click File Menu 2) Click Exit	N/A	Program quits	Program quits
Check graph and Goals Update	1) Click the Food Diary menu 2) click add a new food button 3) enter name 4) enter weight 5) enter calories 6) enter protein 7) enter carbs 8) enter fat 9)click save the above as a new food button 10) click the home menu	Food Name: Bacon Food weight: 100 Calories: 350 Protein:45 Carbs: 35 Fat: 15	Graph and Goals reflect changes	Graph and Goals reflect changes

Food Diary Page				
Test Scenario	**Test Steps**	**Test Data**	**Expected Results**	**Actual Results**
Change date	1) change date selector	Date: 08 December 2018	Diary view changes to correct day	Diary view changes to correct day
Change date to invalid date	1) change date selector	Date: 31 February 2018	Warning issued	Warning issued
Click on an entry in the diary	1) change date selector 2) click on a diary entry	Date: 08 December 2018	Populated Update interface is shown	Populated Update interface is shown
Delete an entry in the diary but cancel	1) change date selector 2) click on a diary entry 3) click delete this food button 4) Click No when prompted	Date: 08 December 2018	Item remains	Item remains

Delete an entry in the diary	1) change date selector 2) click on a diary entry 3) click delete this food button 4) Click Yes when prompted	Date: 08 December 2018	Item is deleted	Item is deleted
Update and entry in the diary	1) change date selector 2) click on a diary entry 3) Edit the data 4) Click update this food button	Date: 08 December 2018	Item is updated	Item is updated
Show the insert interface	1) click on add a new food button	N/A	Blank Insert interface is shown and save button	Blank Insert interface is shown and save button
Insert a new entry in the diary	1) click on add a new food button 2) enter name 3) enter weight 4) enter calories 5) enter protein 6) enter carbs 7) enter fat 8)click save the above as a new food button	Food Name: Bacon Food weight: 100 Calories: 350 Protein:45 Carbs: 35 Fat: 15	Item is created	Item is created
Backup the Diary	1) click on The Backup Diary button	N/A	File is created in the current directory the format username_date.bak (test_10122018.bak)	File is created in the current directory the format username_date.bak (test_10122018.bak)

5 Critique

Having prior programming experience I wanted to challenge myself a little with the project and chose to use tkinter GUI and a database backend. Tkinter is a totally new learning experience and it was a steep learning curve, however I am pleased with the outcome overall. The application functions as expected and all the core functionality was achieved. There was time to include some of the additional features such as the graph and colour coded goals and an export feature.

Not all the additional features were implemented. Calculating macros based on height and weight remains uncompleted as does an import feature. Both these features are possible. The macros could be calculated on the fly quite easily however, due to time constraints this was not implemented. The import feature is also completely possible but more thought is needed on how this would affect any existing records in the database. A way to remove any duplicate entries would need to be implemented to achieve this.

While developing the application, several possible improvements presented themselves. The most obvious being unit conversion, which would allow a user to enter their weight and height and the weight of their food in any measurements, calculating the relevant conversions for storage behind the scenes.

It would also be possible to validate macronutrient totals by calculating values of 4 calories per gram of protein and carbohydrate and 9 calories per gram of fat. This would allow for validation of the goals when a user sets up an account or edits their goals and would stop accidental increases in values which may remain unnoticed for some time.

With regards to security, the passwords are stored unhashed in the database, using a hashing algorithm it would be possible to hash these before saving the record, meaning that if anyone opens the database they would be unable to read the passwords.

The backup feature also stores the backup in an unencrypted format and this should be encrypted to allow for greater security. The backup could be further improved by allowing for a date range when doing a backup instead of backing up the entire diary.