**CT4021 Introduction to Programming Fundamentals**

**Assignment 001**

Technical Report

for the Expense Management Program

**Student Name**: Nathaniel Clarke

**Student Number**: s1906563

**Course**: Computer and Cyber Security

**Submission Date**: Thursday, 16 January 2020

**University of Gloucestershire**

Contents

[Introduction 3](#_Toc29977684)

[Design 4](#_Toc29977685)

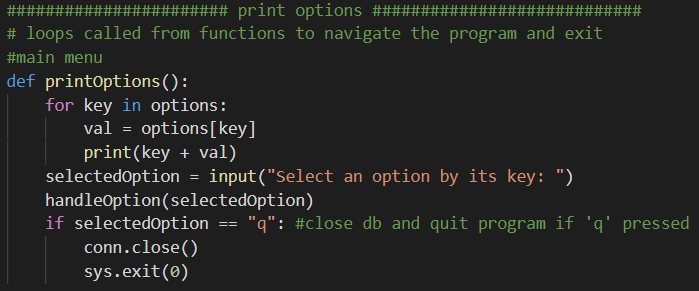
[Testing 5](#_Toc29977686)

[Appendix 2 - User Manual 6](#_Toc29977687)

[References 7](#_Toc29977688)

# Introduction

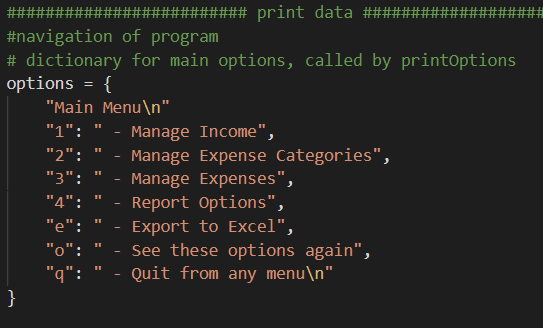
# Design

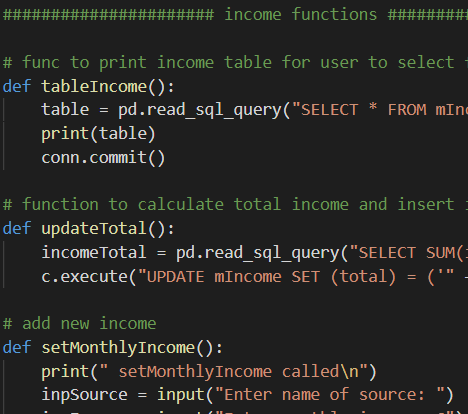
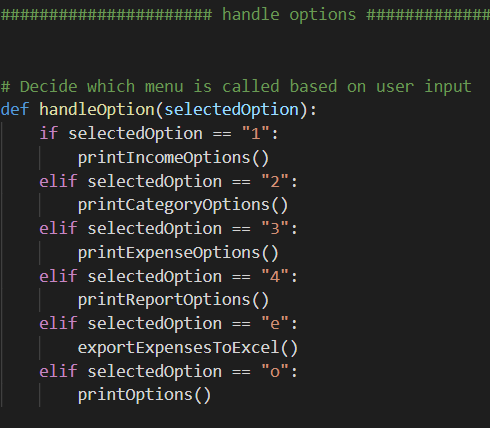
****

The software has been designed so that users can easily navigate the program and quickly access its functions.

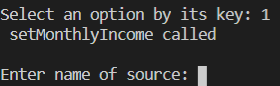
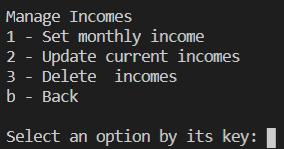
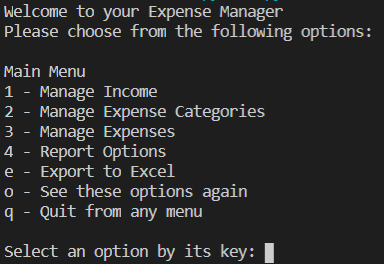
The code has been built using dictionaries for the menu that get called by a corresponding function. When the program is first started the main menu function calls the options dictionary and, based on user input, the next print function is called. Functions that are in the same category or perform the same function are grouped together with title in a line of “#”s to distinguish. This keeps the program tidy and helps the programmer or others to easily understand what’s going on.

Every task the software must execute has been put into a function to be easily called upon. The handleOption functions near the top of the code decide what task is called based on the user input.

****



This is what all that code coming together looks like in the terminal



# Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ref.** | **Scope** | **Expected Results** | **Actual Results** | **Fixes** |
| 001 | Program start   1. Correct input 2. Incorrect input | Main menu brought up, input takes user to corresponding menu, incorrect input just loops to make input again. User can navigate through menus | As expected | N/A |
| 002 | Quit program from any menu   1. ‘q’ input in menu | Database closed, program ended | As expected | N/A |
| 003 | Set monthly income   1. Input source name 2. Input income | Values saved to dB, total income calculated and updated, user alerted/brought back to menu | As expected | N/A |
| 004 | Add new category   1. Input category name | Category saved to dB, user alerted/brought back to menu | As expected | N/A |
| 005 | Add new expense   1. Input expense name 2. Input category 3. Input cost 4. Input date | If category input matches existing category - add expense to dB and calculate then update over/under for expense and category  If category input invalid - user alerted, back to menu | To add over/under Numpy float could not be concatenated as integer | Dataframe value had to be converted to string first |
| 006 | Update income   1. Select source to change 2. Enter new values | If input matches existing - old values replaced with new ones in dB, user alerted  If input invalid – user alerted, back to menu | Error inserting into dB | Was trying to use INSERT with WHERE clause instead of UPDATE |
| 007 | Update category   1. Select category to change 2. Input new name | If input matches existing category – update category name in dB  If input invalid – user alerted, back to menu | As expected | N/A |
| 008 | Update expenses   1. Select expense to change 2. Input new name, category, cost | If expense and category input match existing – update values  If expense or category input invalid – user alerted, back to menu | Over/under calculation values undefined | Sql execute query had to be above calculation logic |
| 009 | Delete income, category, expense   1. In according function, select income, category or expense to delete | If income, category or expense input exists – delete target entry from dB table  If input invalid – user alerted, back to menu | Delete expense did not update catTotal | Selecting only cost of 1 expense, not whole category. Changed select query to get sum of all expenses in category |
| 010 | Export to excel   1. Input ‘e’ in main menu | All dB tables exported nicely into excel sheet, user alerted and taken back to menu | Data exported but tables overlapping | Start each table with a gap between last |
| 011 | Generate expense report of all expenses   1. Option 1 in report menu | Pdf report generated in reports folder with name ‘All Expenses’, user alerted and taken back to menu | Error finding directory | Removed ‘/’ before reports in pdf named |
| 012 | Generate expense report by date   1. Select view by range or day 2. If by range – input dates to find expenses between 3. By day – input 1 date | Report generated in reports folder with dates in title, all expenses between specified dates or from specified day shown in bar chart, user alerted and taken back to menu | As Expected | N/A |
| 013 | Generate expense report by category   1. Input category | Report generated in reports folder with category in title, all expenses under category shown in bar chart, user alerted and taken back to menu | As expected | N/A |
| 014 | Generate report of over/under for category   1. Input category 2. Input dates to view between | Report generated in reports folder with category in title for over/under, all expenses matching category and date parameters shown in bar chart | Only one expense being obtained from dB | Changed ‘fetchone’ to ‘fetchall’ in sql query |

# Conclusion

To improve the software next time I would:

1. Have better validation on the date inputs, can easily still be incorrect
2. Create functions with different parameters rather than several functions that do slightly different things, more efficient
3. Add a line for over/under in the other reports instead of whole new report

# References

Check exists

<https://stackoverflow.com/questions/43246384/pythonsqlite3-if-statement-to-see-if-value-is-in-database>

SQLite db

<https://www.sqlitetutorial.net/sqlite-python/creating-database/>

<https://www.sqlitetutorial.net/sqlite-python/?fbclid=IwAR1D-QLVuo6-sUxtf8IaqLqEtTvOKNKsvfdq739s6dWdxcLm5OFvRWuUlp8>

Dictionary

<https://www.w3schools.com/python/python_dictionaries.asp>

Pandas print table

<https://stackoverflow.com/questions/37051516/printing-a-properly-formatted-sqlite-table-in-python>

XlsxWriter

<https://xlsxwriter.readthedocs.io/example_pandas_multiple.html>

<https://stackoverflow.com/questions/32957441/putting-many-python-pandas-dataframes-to-one-excel-worksheet>

# Appendix 1 - User Manual