

# **CS-A1123 Y2 Final Project**

## **Money Management**

**Cao My Nhut - 906939**

Bachelor of Technology and Science – Digital System and Design

May 11, 2023

## **2. General description**

This application is created for the final project of the course CS-A1123. Choosing Money Management as a topic, the creator has made use of Python and PyQt6 library to create the application.

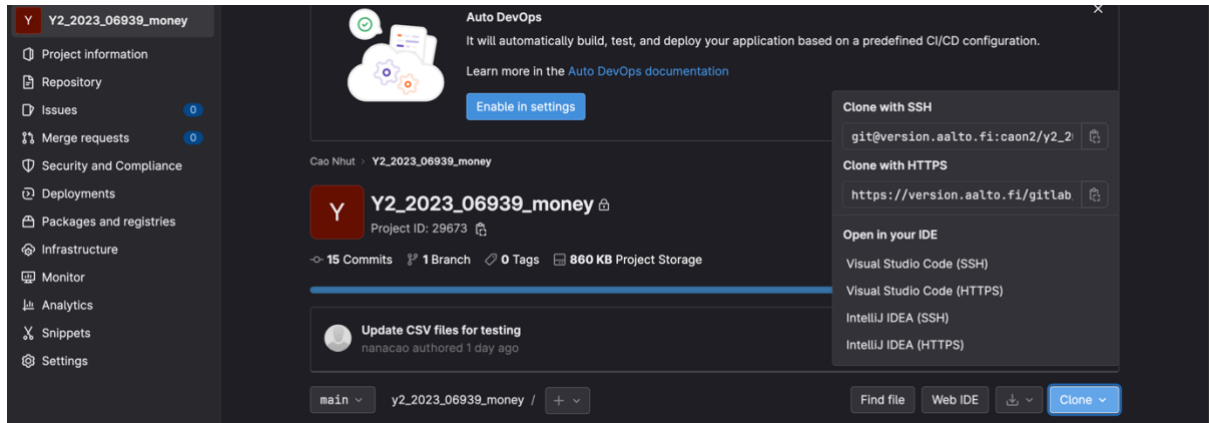
With this, user can upload CSV files that contain information such as amount of money, date, receiver, or sender to the app and get a pie chart regarding the expense or income. Other than that, user will have a table where all the expenses are listed, and based on a certain threshold, it will be displayed in red to mark as important.

This version of Money Management has managed to fulfill the medium level of the course requirements. Initially aim for the hard level, the creator, however, can't debug the saving algorithm in the given time. Moreover, according to the plan, the application would have a welcome window. But the creator had to adjust to their ability so the application will start directly with the main menu window.

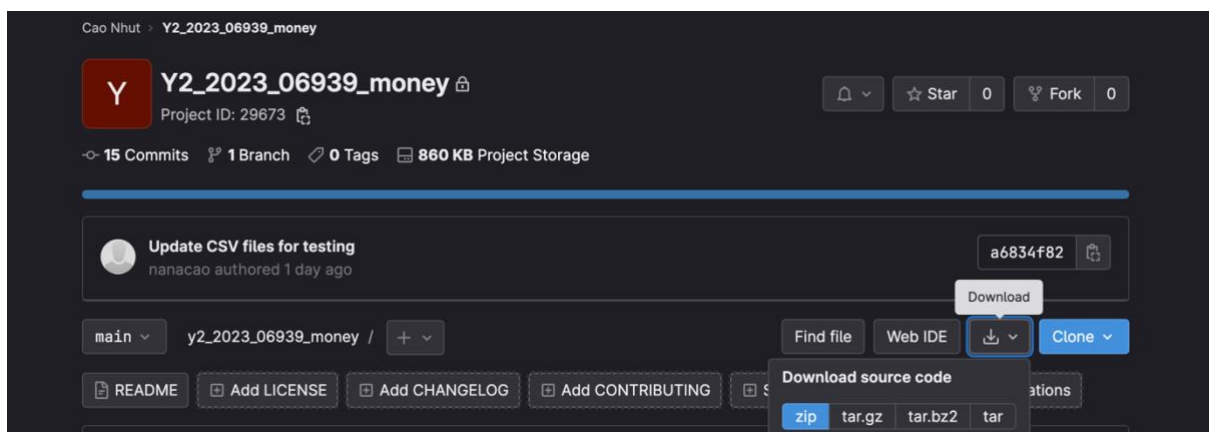
This document contains 15 sections, including general description, instructions for user, external libraries, structure of the program, algorithms, data structures, files, testing, the known shortcomings and flaws in the program, three best and three worst areas, changes to the original plan, realized order and scheduled, assessment of the final result, references, and attachments.

### 3. Instruction for users

From the GitLab repository, user can clone the project given the user has setting up SSH keys and match with GitLab. Otherwise, users can directly download the whole package.



*(Clone with SSH or HTTPS, if user comfortable with Git)*



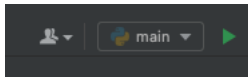
*(Download the whole package)*

README file has partially show how to run the program. For more specific, user once have the program folder locally, kindly navigate to it using command line or terminal, or user can open the folder with an IDE of choice e.g. PyCharm. Note: Users make sure to have Python version 3.x installed first.

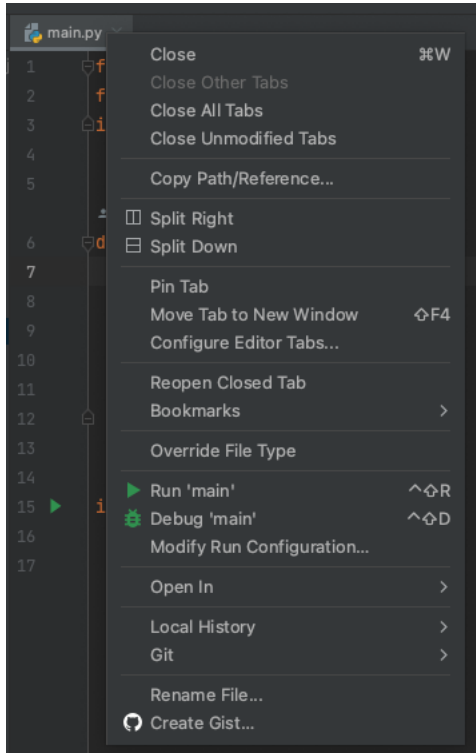
- With command line, once users are in the correct directory (starts with something like `y2_2023_06939_money`), users can use command “`ls`” to check if all the files are presented. The source code is stored in a sub-folder called “`src`” so users need to navigate to that folder by command “`cd src`”, then run the program with “`python3 main.py`”. The main window of the application will be displayed

```
y2_2023_06939_money ~  
y2_2023_06939_money ~ cd src  
src ~ python3 main.py
```

- With an IDE, here the creator uses PyCharm, users can easily run by clicking a green button in the opening main.py file



Or can right click at the name of the file and select **Run 'main'** as below



Once the users have the main window opened like below, they can upload their CSV files in the given field and start the program with the GUI easily (buttons, descriptions provided).

Other Operations

Income

Expenses

Checked €: 0.0 (0.0%)

Savings €: 0.0

1) Drag and drop bank files in csv format to below window to graph a pie chart.  
2) Switch between income and expenses graphs by clicking above buttons "Income" and "Expenses".  
3) Check expenses checkboxes on the right to group expenses rows and set their importance:  
- Rows in red are set as important expenses.  
- The program automatically sets most frequent and sum wise biggest expenses as important.  
- Use unset importance button to set expenses row as unimportant.  
- In order to group, write title of the new group and press " Group" button.  
- Expenses row can also be just renamed by checking only one row.  
4) Click "Other operations" from above menu bar in order to do additional operations:  
- Choose "Clear window" to clear loaded bank files and graphs.  
- Choose "Show info" to display this info text whenever it is needed.  
- Choose "Delete row" to delete specific rows from the left.  
- Choose "Quit" to exit the program.  
5) Graph a savings pie chart by writing savings amount and pressing "Save Money":  
- The graph will show suggested expenses amounts based on importance settings of expenses rows.  
- Savings amounts will not be applied to important expenses.

Load file

Set importance

Save money

Group

Welcome to MyMoney Manager.

Unset importance

Savings amount

Group title

Instructions for using MyMoney Manager

Users can upload as many files as they want into the field, then they can select which file they want to see the statistic by click the file. If it is highlight in red, then users can press “Load file” to execute the program. First, the expense pie chart will be shown alongside with the rows where all the expenses are listed. Users can click “Income button” in upper screen, then the income pie chart is shown.

With “Group” button, users can choose several rows of expenses to group, for example Expenses for grocery. With “Set importance” and “Unset importance”, users can choose to set if chosen expenses/income is important or not. Usually, the program set the limit of 500 euros for income to be important, and expense that greater than or equal to 100 to be important.

There is also an option called “Other operations” where users can choose whether to clear the window, delete a row, show the info table again, or close the program completely.

## 4. External Libraries

The application makes use of PyQt6 as much as possible, and with some help from other common library such as “*threading*”, “*sys*”, “*random*”. No other external libraries are used to build the application.

## 5. Structure of the program

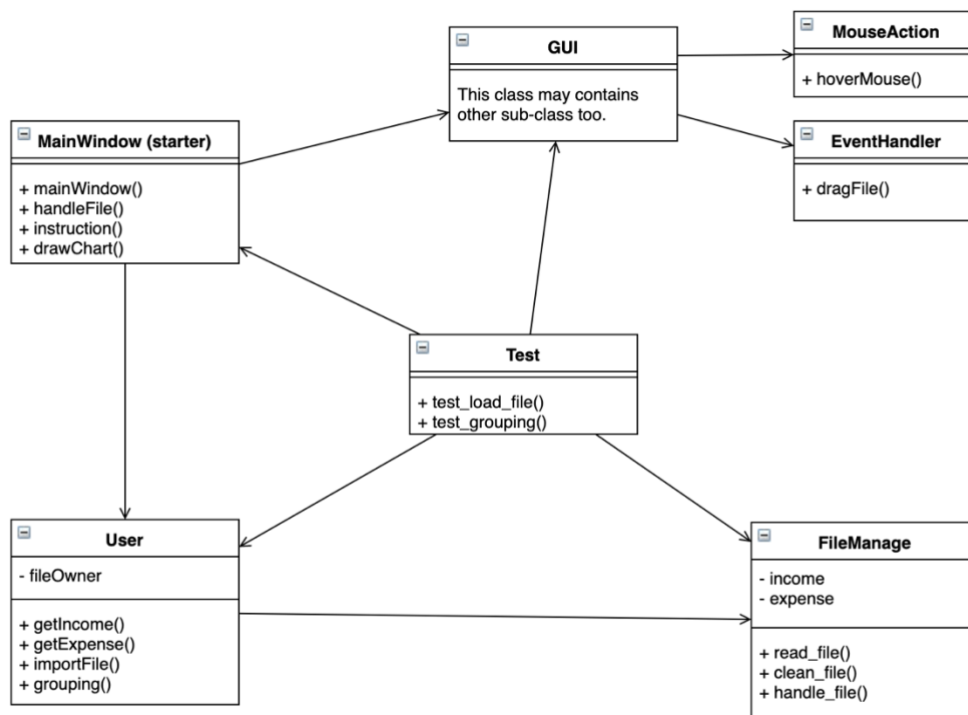
The program has 6 classes in total, and MainWindow class is the most crucial class of all. In this class, the functionalities and GUI display are implemented. Important variables, instances, methods are mostly located in MainWindow class. The methods are name in a way that reader can understand what they are used for. Some central methods such as *delete\_rows*, *group\_expenses*, label and buttons create methods, *create\_info*, *create\_menu\_bar* where the information is displayed. These methods closely associated with class instances and methods come from other classes as well. Pie charts create methods (*create\_expenses\_piechart*, *create\_income\_piechart*) are also located in this class, where income pie chart and expenses pie chart are formulated. The methods heavily based on the library PyQt6 and its properties.

Class User handles the grouping, set importance/unimportance functions, as well as delete row. It also cooperates with class FileManager to support its methods.

Class FileManager handles import and read file functions (*read\_file*). It has methods (*organize\_rows*) to re-structure the file to sanitize unwanted row and format the file data into a suitable format for other methods to continue. This class also offers methods for class User to implement the grouping, setting importance/unimportance options.

Class ListBoxWidget, EllipseItem, and LineEditWidget are mainly used for implementing the GUI, which class MainWindow will take advantage of. These three classes use several of methods from PyQt6.

Below is the UML diagram



## 6. Algorithms

The program contains very simple and basic algorithm, as the hard level is not satisfied, the creator hasn't managed to implement advance algorithm to support these requirements. For this application, looping is widely used. Assigning values to variables are similar to what students learn from the course.

The creator didn't apply advanced mathematical computation so there is no formula. The creator believe there exists a more concrete, shorter version of implementation; however, this application is by far what the creator can do with her ability.

## 7. Data structures

The creator used dictionary as the main data structure for storage, beside list. With key-value relationship, it is more convenient to manipulate the data. List may also satisfy the requirements, but then sub-lists will be involved and require more complex implementation and effort which are not necessary as we can do it with dictionary. Immutable data structure such as tuple is not widely used since the program can not predict the users' actions so it should be flexible to handle all the request.

## 8. Files

The program can handle only CSV files, as it is the most common type that banks provide. The creator also wanted the program can handle more types of file but the time and ability are limited. When users download or clone the code, they are given 3 CSV files that are flawless, some blank

lines, and faulty file to try out the application or to run the test. CSV files that the program can read usually have 8 columns, included *date*, *amount*, *sender*, *receiver*, *name*, *title*, *reference number*, *currency*. This format is originally from the bank, and when the program read the files it will only process the necessary columns and rows.

## 9. Testing

The test file is in the “src” folder and has the same level with the CSV files too. There are total 10 tests, which are for testing reading file (flawless, blank line, faulty file), test the functionalities, test reading multiple files at the same time with the functionalities, test a big, whole program.

Users can run all the test by clicking the same button as users click to run the main file, but in this case it the test.py file. Users can also run separate tests by click the green button in front of each test like below:

```
69 ▶ def test_set_important_2(self): # Test that unsetting importance works properly
```

The program able to pass all the given tests

## 10.The known shortcomings and flaws in the program

The application sums up the expenses and income when users load a new file, which make the pie chart look denser. This problem has happened when the creator first implements the unit tests, however it was fixed by using class variable. The creator is looking into the solution for the pie chart version, and the idea is that the data structures are not handled well in the 2 functions to create pie chart.

The color of the pie chart sometimes is too bright which may make users feel uncomfortable, and the scrolling in the window is not as smooth as expected. The creator should look into the PyQt6 documentation to make sure she has correctly used the methods.

## 11.Three best and three worst areas

Best:

- Simple GUI so users can easily use
- Detail information in the main window with all necessary things
- Arrangement of buttons helps users navigate easier.

Worst:

- Color of the GUI is not well cared, especially the pie charts
- The application keeps summing up the money instead of separating.
- Saving algorithm is not working as expected, and the creator is still trying to debug.

## 12.Changes to original plan

The creator aims to have a welcome window before entering to the main window but changed to display the main window instead due to creator’s ability and time source.

The hard level is not satisfied, but the creator decided to keep the saving algorithm code to examine further. So this application meets the medium requirement.

The implementation order followed the plan.

## 13.Realized order and scheduled

The program first started with class MainWindow and the main.py file to testing out the PyQt6 library. Eventually the class FileManager was added with its methods and then class User. Class MainWindow then used the methods from these classes too. There are 3 GUI classes that were added and support the GUI setting in class MainWindow.

Test class was implemented when the handle files functions were completed, initially there was 2 tests to test the functionality.

The program was implemented according the scheduled stated in the plan.

## **14. Assessments of the final result**

As stated in the above sections, the program still not very matured, and has a lot more to improve. The saving algorithm is not working so it is certainly a flaw in the application. The creator can continue to spend more time to improve the algorithm. Moreover, the creator can find a shorter way to implement some methods and make use of more Python data structures and PyQt6 properties.

Classes of the program are well organized. Names of variables and methods are easy to read and understand, therefore users can also know what the code is doing. The GUI of the application is simple to use, help users not feeling overwhelmed and lost when first use the application.

The program can be expanded to have more functions and handles more requests from users, as its structure is very easy to read and add more methods.

## **15. References**

StackOverflow for debugging common Python syntax

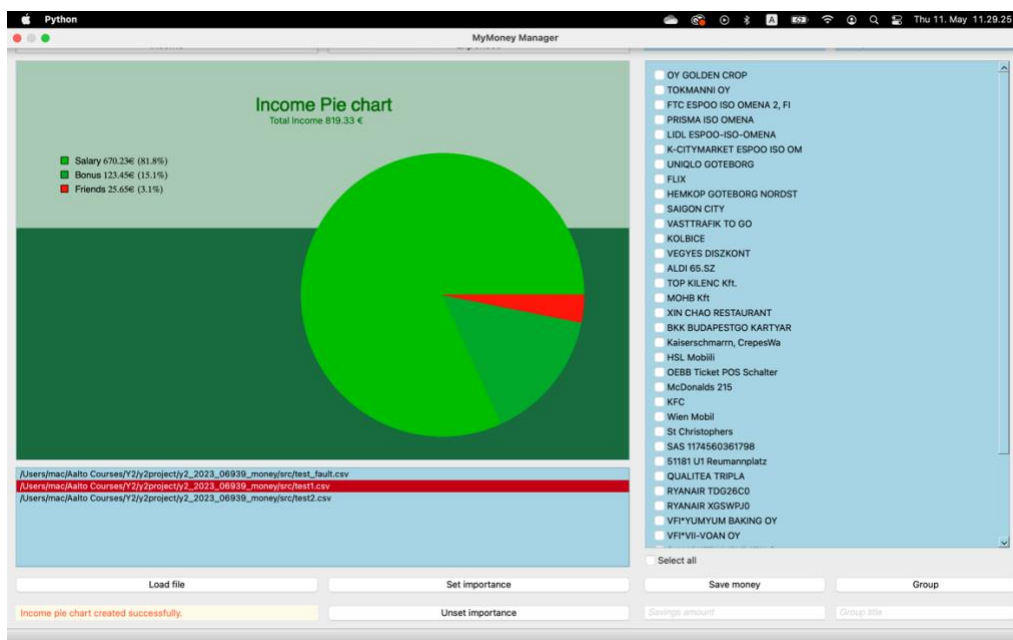
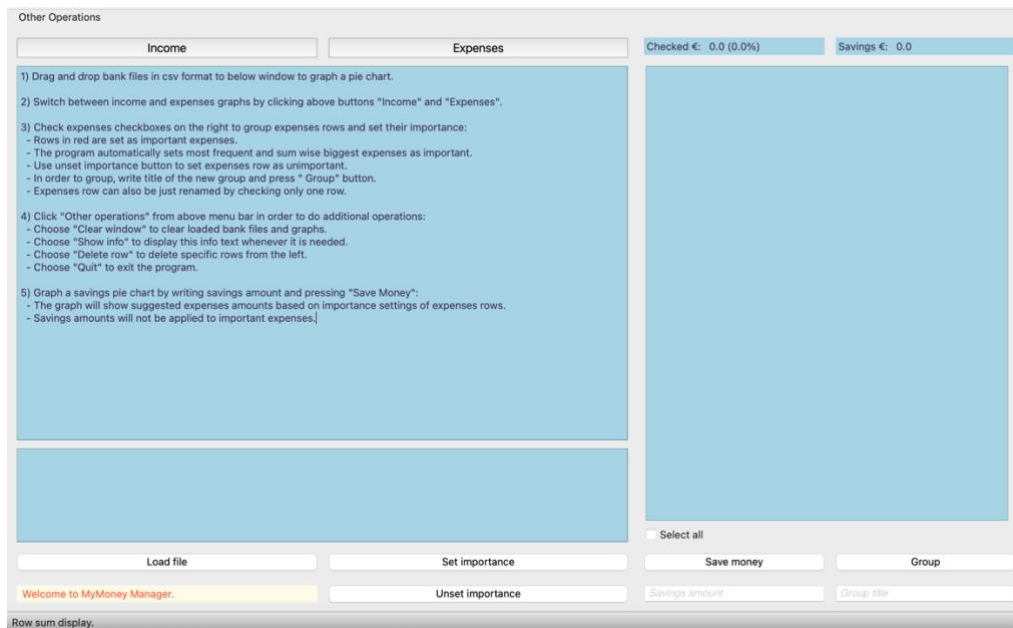
PyQt documentation <https://doc.qt.io/qt.html>

CS-A1123 and CS-A1113 Programming Course Y2/Y1 materials are utilized

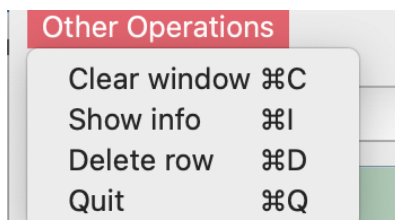
## **16. Attachments**

Users can drag their CSV files to the field, click the file then click “Load file” then the pie charts and information will be displayed.





Other operations options (Command for Mac, Ctrl for Windows users)



To sum up, thanks for reading this documentation and spending time try out the application. All opinions are welcomed so that the creator can improve the application. This is a student-built application and serve no commercial purpose, therefore it may look raw, not professional. The creator hopes users will have a fun time trying out this program.