

**Project documentation**

Subject: **Programming languages**

Project title: **Moneyer**

Teacher: Student:

*Dr Marek Jaszuk Artur Lubiv*

Index no: 59070

Rzeszów 2019

The documentation contents:

* Specification – the main assumption about the goal of the project and it’s functionality.
* The implementation – describe how the project was implemented
* User guide – demonstrate how to use the application
* Source materials that were used while doing the project – specify all the materials that were helpful for creating the application (tutorials, books, video courses, etc.)

Specification

Moneyer is the utility to keep control over your finances. It will compile your daily money flow and show it to you in a balance table over a current month. In the end of the month every table will be saved to history and used in the calculations for the current balance value. The money flow is tracked by regular record of all transactions. Monthly balance table shows your financial history on both income and expense sides. Your current balance is shown and dynamically changes as you earn or spend.

The Implementation

The graphic user interface is implemented using PyQT. It consists of the dialog for initial balance, main window and the transaction creation dialog.

In the initial dialog we have a QDoubleSpinBox for user to provide his available funds.

In the main window we have the balance table laid out on a QGridLayout beginning with the Initial balance displayed. Its divided onto two sides, Income and Expense. On each side there are records of the transactions, each laid out in the separate QHBoxLayout consisting three QLabels displaying the regarding data. Below each side there are QLables displaying total values. On the bottom of the layout is a QLabel displaying the final balance. In the very bottom of the main window is a QPushButton called Add Transaction that calls the QDialog.

In the transaction creation dialog, we have the set of fields:

- QLineEdit to receive the title of the transaction

- QDoubleSpinBox for the amount of the transaction

- QRadioButtons for deciding whether transaction is an expense or income

- QDateTimeEdit for getting the time of the transaction

They are laid out vertically each in the pair with the corresponding QLabel in the separate QHBoxLayout. On the bottom there is QDialogButtonBox for accepting the transaction.

The code part generates the balance class that handles the pandas DataFrame and is able to save it to the csv file, load it and append it with the transaction. Transaction class keeps the information about the single transaction (its title, amount, type, date and time) and can turn it into a dictionary to be appended into the balance.

Main function runs the QApplication as well as generating the current balance content of which is depending on weather it was generate before. If wasn't the Beginning window is shown for user to provide the initial value for the first balance. After, the main window is shown and refreshed.

refresh() cleans the outdated layout, loops throughout the balance rows, creating the QHBoxLayouts containing the QLables for the transaction data, then it decides on which side of the QGridLayout to put it, adds the value to its side's total and is being inserted in the proper cell. Outside of the loop it calculates the final balance adding the initial balance to the total of the income and substitutes the total of the expense. This method is also called when new transaction was added.

User Guide

When you first enter the app, you need to provide the amount of the money you have. Then, you will see it in the main menu. Press Add Transaction and you will see a set of fields to fill. Provide the needed information about the transaction and click OK. You will see your transaction in the table. Whenever you get or spend your money, you need to create the corresponding transaction. Under each of the sides you can see the totals of how much you have spent and earned. On the bottom is your current balance which should be the same as the amount of money you have.

Source materials

* Programming languages presentations
* Stackoverflow.com
* Doc.qt.io
* [www.datacamp.com](http://www.datacamp.com/)
* docs.python.org
* [www.w3schools.com](http://www.w3schools.com/)
* codeyarns.com
* effbot.org
* github.com
* therenegadecoder.com
* [www.learnpython.org](http://www.learnpython.org/)
* [www.riverbankcomputing.com](http://www.riverbankcomputing.com/)
* [http://zetcode.com](http://zetcode.com/)
* codereview.stackexchange.com
* forum.qt.io
* [www.qtcentre.org](http://www.qtcentre.org/)
* data36.com
* towardsdatascience.com
* [www.geeksforgeeks.org](http://www.geeksforgeeks.org/)
* wiki.python.org
* realpython.com
* [www.codementor.io](http://www.codementor.io/)
* [www.digitalocean.com](http://www.digitalocean.com/)