

# Home assignment 2

## Introduction

In the second home assignment we will use the original excel file from the first home assignment. We add a button that shows a user form describing the very basic functions of a crypto portfolio. User can add a new transaction and load a history of existing transactions.

## First steps

First add a button on the existing excel sheet. It will be something like that:

Currency name	Created date	MarketCap	Price on 01 Jan	Price on 01 January 2015	Price on 01 January 2016	Price on 01 January 2017	Price on 01 January 2018	Price on 01 January 2019	Age in days	Age in "years months days" format
Bitcoin	1/9/2009	\$64,072,742,282.00	754.97	320.44	430.72	963.66	14112.2	3746.71	3700	10 years 1 months 17 days
XRP	1/1/2012	\$12,765,840,835.00	0.027365	0.024455	0.00604	0.006523	2.3	0.352512	2613	7 years 1 months 25 days
Ethereum	7/30/2015	\$12,468,721,197.00			0.933712	7.98	755.76	132.87	1307	3 years 6 months 27 days
Litecoin	10/12/2011	\$2,633,424,831.00	24.35	2.72	3.48	4.33	231.67	30.46	2694	7 years 4 months 14 days
EOS	1/21/2018	\$2,499,390,343.00					8.77	2.56	391	1 years 0 months 26 days
Bitcoin Cash	8/1/2017	\$2,254,724,782.00					2534.82	150.9	574	1 years 6 months 25 days
Tether	2/15/2015	\$2,025,178,699.00			1	1	1.01	1.01	1472	4 years 0 months 11 days
TRON	9/13/2017	\$1,765,527,759.00					0.044682	0.018767	531	1 years 5 months 21 days
Stellar	8/5/2014	\$1,544,207,485.00		0.005517	0.001752	0.002464	0.360422	0.112411	1666	4 years 6 months 21 days
Binance Coin	7/25/2017	\$1,233,202,051.00					8.63	6.19	581	1 years 7 months 1 days
Total capitalization:		\$103,260,960,264.00								

Currency name	Diff 2014 - 2015	% from the price 2014	Diff 2015 - 2016	% from the price 2015	Diff 2016 - 2017	% from the price 2016	Diff 2017 - 2018	% from the price 2017	Diff 2018 - 2019	% from the price 2018
Bitcoin	-434.53	-57.56%	110.28	34.42%	532.94	123.73%	13148.54	1364.44%	-10365.49	-73.45%
XRP	-0.00291	-10.63%	-0.018415	-75.30%	0.000483	8.00%	2.293477	35159.85%	-1.947488	-84.67%
Ethereum	0	0.00%	0.933712	0.00%	7.046288	754.65%	747.78	9370.68%	-622.89	-82.42%
Litecoin	-21.63	-88.83%	0.76	27.94%	0.85	24.43%	227.34	5250.35%	-201.21	-86.85%
EOS	0	0.00%	0	0.00%	0	0.00%	0	0.00%	-6.21	-70.81%
Bitcoin Cash	0	0.00%	0	0.00%	0	0.00%	0	0.00%	-2383.92	-94.05%
Tether	0	0.00%	0	0.00%	0	0.00%	0.01	1.00%	0	0.00%
TRON	0	0.00%	0	0.00%	0	0.00%	0	0.00%	-0.025915	-58.06%
Stellar	0	0.00%	-0.003765	-68.24%	0.000712	40.64%	0.357958	14527.52%	-0.248011	-68.81%
Binance Coin	0	0.00%	0	0.00%	0	0.00%	0	0.00%	-2.44	-28.27%

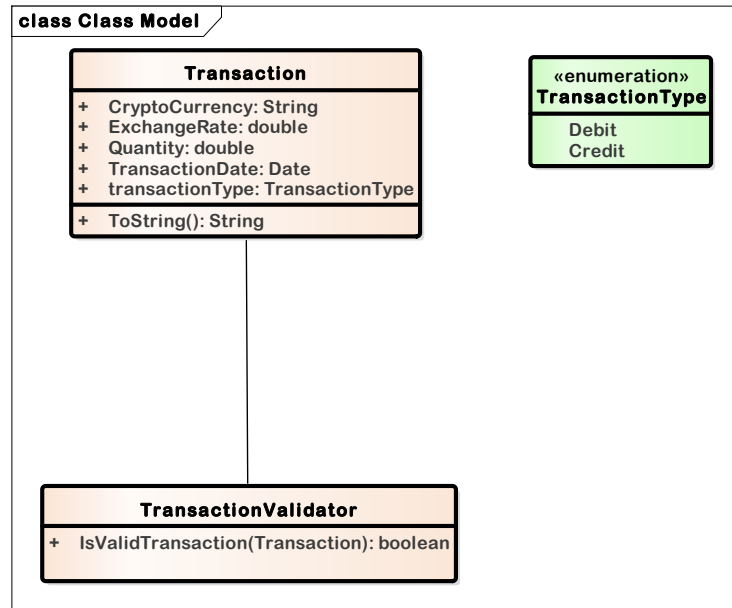
The information about most capitalized currency.  
The most capitalized crypto is XRP with capitalization of 64072742282 dollars. The XRP was created on 01.01.2012

Portfolio

Figure 1 New button for opening a form

## Class diagram

Class diagram describes the main classes and enumerations in VBA code. For our system we need one class for transaction itself (*Transaction*) and transaction validator to validate the user input (*TransactionValidator*). Transactions can be both debit (if user buys something) and credit (if user sells something).



*Figure 2 Class diagram for VBA code*

The transaction has public fields for the name of the currency, its quantity, exchange rate, date and type. For example, if user wants to add a transaction that he bought 2 bitcoins with a price of 3500 USD each we will have a transaction object with fields:

- Transaction
  - `CryptoCurrency` = "Bitcoin"
  - `ExchangeRate` = 3500
  - `Quantity` = 2
  - `TransactionType` = Debit
  - `TransactionDate` = current date (just fill it automatically to current date)

Transaction validator should check that

- Transaction object exists (Not Nothing)
- `CryptoCurrency` is filled (not an empty string)
- `Quantity` is more than 0
- `Exchange rate` is more than 0

If all these conditions are met, then validator will return true and transaction can be saved otherwise show message box saying something like "Please enter correct data."

## Use case models

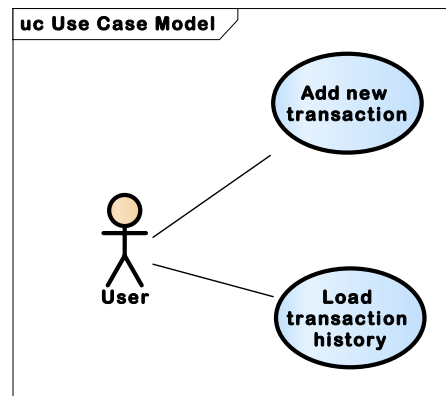


Figure 3 Main user functions

As shown on the use case diagram user can have 2 functions available:

1. Add new transaction
2. Load transaction history

### Add transaction

When user adds a transaction it is being validated and if valid then stored in the excel sheet called "Transactions" (or some other name). For entering transaction details please use a form similar to:

The screenshot shows a software window titled "Portfolio" with a close button (X) in the top right corner. The window is divided into two main sections. The top section, titled "Transaction history", contains a large, empty rectangular area. The bottom section contains input fields and buttons. On the left, there is a "Currency" dropdown menu set to "Bitcoin", a "Quantity" text box with the value "2", and an "Exchange rate" text box with the value "3500". Below these is a "Buy or Sell" section with two radio buttons, "Buy" (selected) and "Sell". At the bottom left is an "Add transaction" button. To the right of this are two text boxes labeled "Total debit" and "Total credit". At the bottom right is a "Load history" button.

Figure 4 Example of a user form

On the left side you can see different GUI components for entering transaction details:

1. Combobox for currency name. Currencies should be taken from the original sheet (A11 : A20). To fill the ComboBox use its property *RowSource*
2. Input for Quantity (TextBox)
3. Input for Exchange rate (TextBox)
4. Frame for Buy or Sell options with radio groups
5. Button "Add transaction"

To save transaction, first create the second sheet similar to:

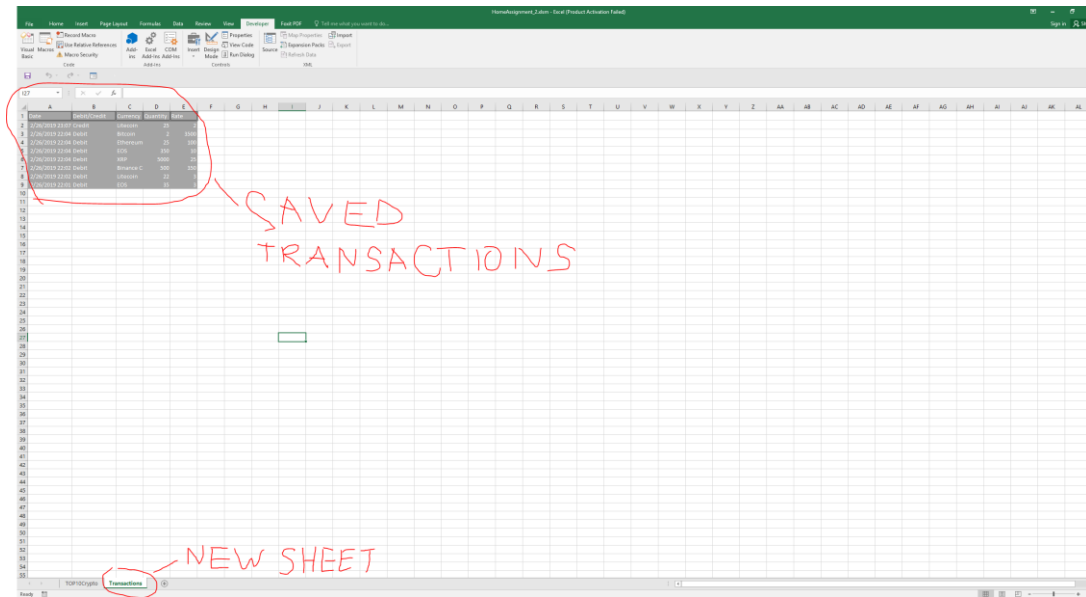


Figure 5 New Excel sheet for storing transactions

Please note that new transaction is added always on the second row (A2: E2). To keep existing transactions and not to override them with a new one shift existing ones down with

`Worksheets("Transactions").Rows(2).Insert Shift:=xlDown`

After that you can add your new data to the second row.

### Load transaction history

To load transaction history you need to read them from the "Transactions" sheet starting from the second row (A2 : E2) until the first empty row. Just check A column and if it is empty then stop reading.

To simplify showing transactions in a ListBox implement method ToString() in Transaction class, so your transaction history will look similar to:

Portfolio

Transaction history

Currency

Quantity

Exchange rate

Buy or Sell

☐ Buy

☐ Sell

Add transaction

26.02.2019 Credit Litecoin quantity: 25 rate: 2

26.02.2019 Debit Bitcoin quantity: 2 rate: 3500

26.02.2019 Debit Ethereum quantity: 25 rate: 100

26.02.2019 Debit EOS quantity: 350 rate: 10

26.02.2019 Debit XRP quantity: 5000 rate: 25

26.02.2019 Debit Binance Coin quantity: 500 rate: 350

26.02.2019 Debit Litecoin quantity: 22 rate: 3

26.02.2019 Debit EOS quantity: 35 rate: 3

Total debit

Total credit

Load history

Figure 6 Transaction history list

ToString() method return a string that is a concatenation of transaction fields, separated with a whitespace and words like “quantity: ” and “rate: ”. You can select how do you want to show transaction in a list, the main requirement that it is readable.

After transactions are loaded, calculate totals for debit and credit transactions and show them in appropriate inputs *Total Debit* and *Total Credit*.

## Requirements

The successful solution should include:

- Classes
- Enum
- Loops
- IF ELSE structures
- The usage of Collection
- Functions

Also, please provide some short description of your solution either as comments in the code or text inside the excel or separate word file.

## Deadline

30 April 2020