

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:

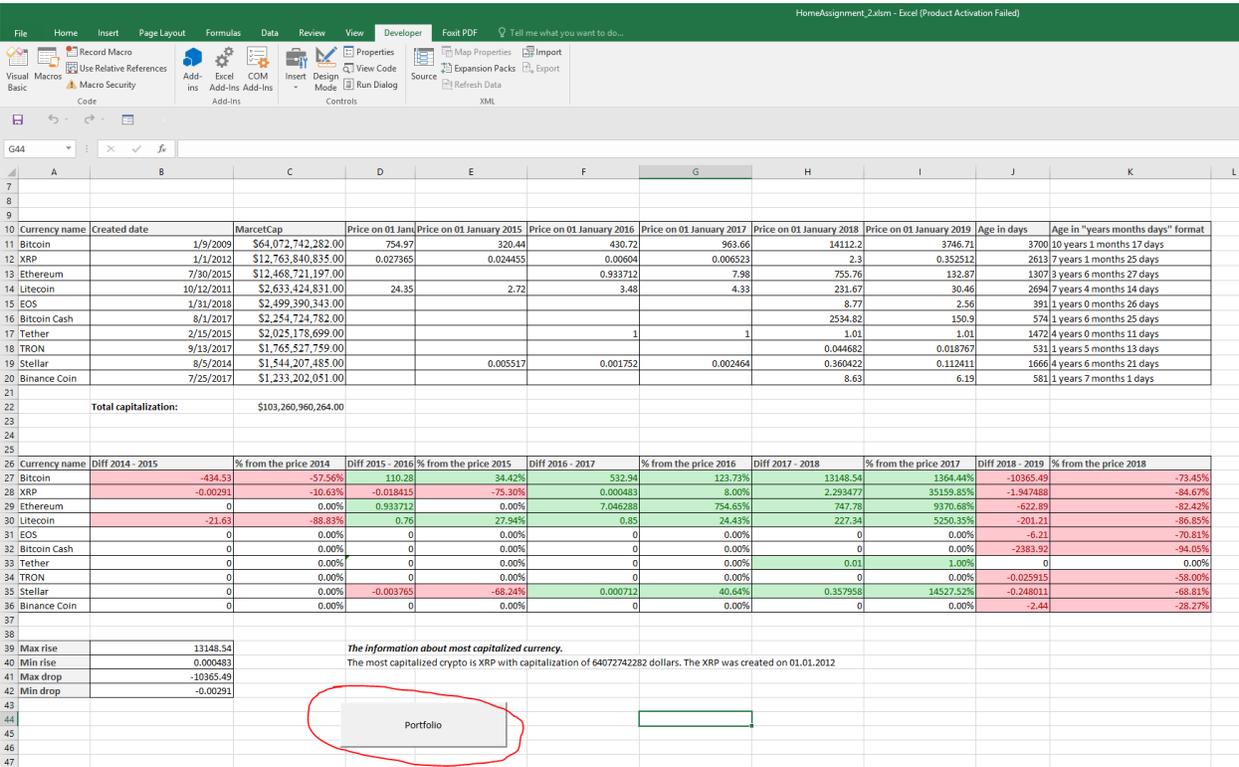


Figure 1 New button for opening a form

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).

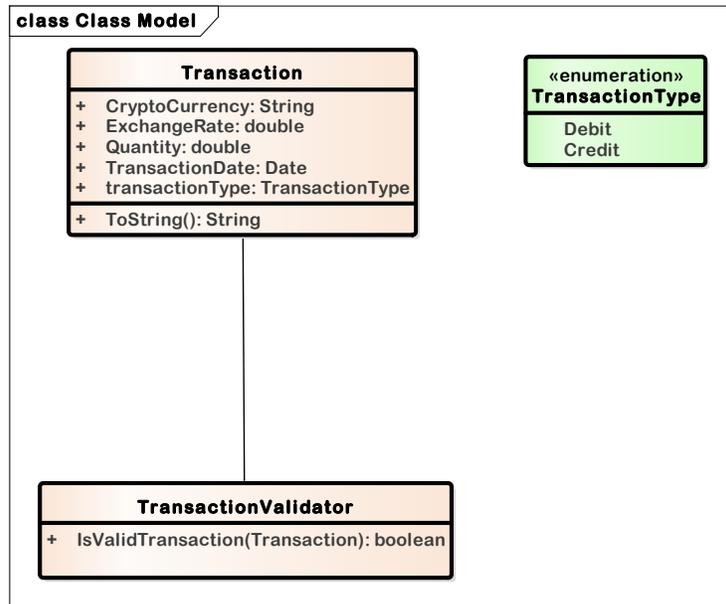


Figure 2 Class diagram for VBA code

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

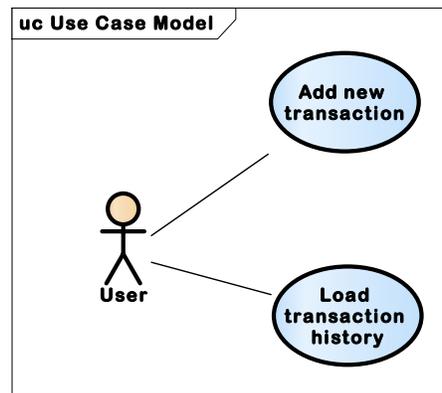


Figure 3 Main user functions

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 window titled "UserForm1" with a "Portfolio" section. On the left, there are input fields: "Currency" with a dropdown menu showing "Bitcoin", "Quantity" with a text box containing "2", and "Exchange rate" with a text box containing "3500". Below these is a "Buy or Sell" section with two radio buttons, "Buy" and "Sell". At the bottom left is an "Add transaction" button. On the right, there is a large empty rectangular area labeled "Transaction history". Below this area are two text boxes for "Total debit" and "Total credit", and 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:

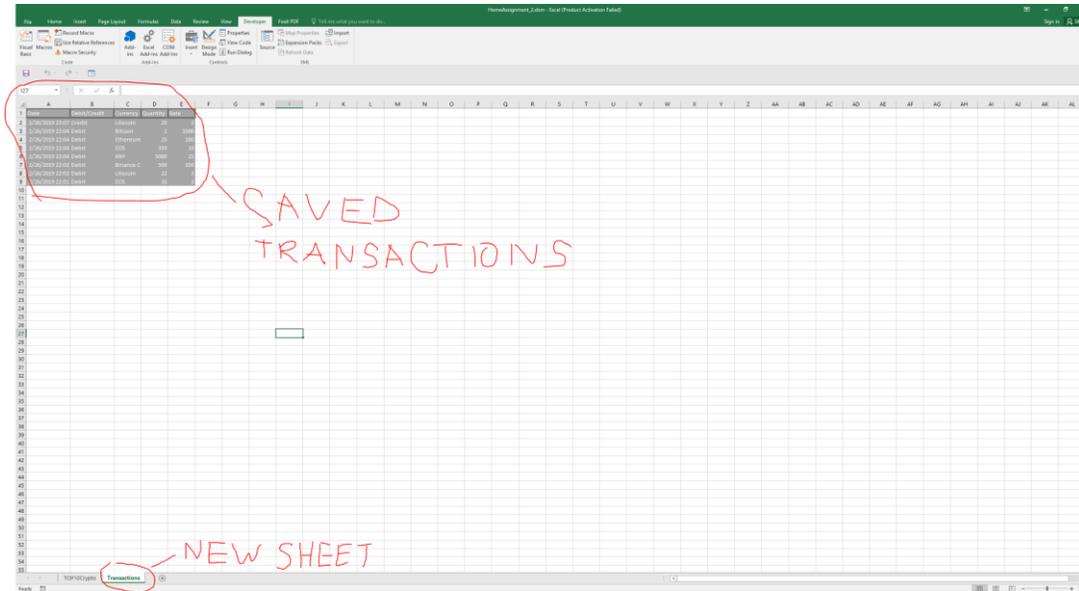


Figure 5 New Excel sheet for storing transactions

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:

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