PROG. DESIGN & CONSTRUCTION

SEM 1 : ASSIGNMENT 2

By Hayden Richard-Marsters

SID: 21152003

Extension to previous part 1: Asset Management System

GitHub: https://github.com/mFiveTrillion/Asset-Management-System

Aim: Create a comprehensive system that provides a subset of tools for managing a user's assets. It allows control over the user's assets variables, such as market values and total amount in holdings. The program aims to create a tool that can be used by anyone holdings multiple assets to keep track of their portfolios profits, costs and position. The program incorporates JDBC (java database connectivity) to maintain data persistence to store various useful information of assets, transactions and overall holdings. The aim was to also create a scalable system to where assets can be controlled at scale, did this by incorporating importing functionality. The user can interact and maintain control over how the portfolio is treated through a generated user interface GUI. Overall, the program aims to create a centralized asset management system that can process scalable amount of processes to manage various amounts of assets from small to large while simultaneously providing distinct data analytics.

Setup: The program consists of 12 distinct classes to provide encapsulation and clear code to the various asset, portfolio, GUI, transaction and database management objects. Through inheritance, Asset includes sub-set classes Stock, Real Estate, and ETF, which all inherit Asset classes' instance variables, constructors, and methods to round the object classes. The portfolio object consists of the portfolios relevant data and methods to pass into metrics calculator which computes the relevant info we need.

Functionalit*y:* the functionality of the program through the GUI consists of 7 sections. Firstly, the import button which imports a list of assets from a text file, checking if the file has been imported or not – the file can only be imported once. Next is the buy and sell asset button, this controls the user wanting to purchase an asset (at time 0), incorporating error handling methods to control the users input and data. The top performing list button returns a sorted list of top performing assets based on the portfolio weighting to net profit / loss ratio. The display portfolio and transaction buttons display the data that populates the asset man DB, the PORTFOLIO table and the TRANSACTIONLIST table

JDBC: The program first when started connects to the database and checks the status of existence of tables within the database, (through hard code the tables can be removed, by uncommenting the call for remove table method) the program automatically checks if the tables are there if they are the program continues, if not the program creates the tables. Finally the system output displays the status of each table, throughout the program the databases are kept up to date in reaction to the relevant methods.