**StockApp Design**

For the Java based Stock Application, we have followed the Model-View- Controller Structure.

Application has 3 components:

1. Model : contains all the logic part
2. Controller : takes input from user and calls appropriate functions in model and view.
3. View : prints all the required statements in the text based interface.

Helper packages and classes:

1. Utils class : this class contains all the function which are happening at file level : reading from a file, saving data in file, check for file and data etc.
2. Apifetcher : this class fetches the data from API and parses the data in a particular function based on the stock name
3. Constants : this package contains all the fixed values like the printing statements, which remain constants.

Controller Design :

* Controller contains the logic to take input from user and display further menu and data based on user’s input
* Controller calls the functions of User Interface which Is the entry point to the model interface.
* It sends the data received from model to the view to print it and also print based on user’s print.
* Since all methods of view are static, we don’t need to create object of view as of now.
* Controller has the object of UserImplementation class and interacts with all methods of model through this object only.

Model Design :

* StockImpl is the lowest level class that Implements Stock interface and has the functionality to get current and historical date value from the local file for a stock. StockImpl has a name, buying date and the buying price of that stock. This class is an immutable class as once a stock is bought, its reference variables shouldn’t be altered.
* StockOrderImpl class is a class on top of Stock which contains the object of Stock along-with its quantity of the stock for a portfolio. This class implements StockOrder interface and returns the total value/pnl of that stock in the platform. This class is also immutable because of the same reason as of StockImpl.
* PortfolioImpl Class represents a portfolio of the user which contains the name and list of stockOrder objects. This class implements Portfolio which contains the functionalities to get value summary, detailedView,

Pnl for that portfolio. This class is immutable as once a portfolio is created, it shouldn’t be modified, just be viewed.

* Model’s class that interacts with the controller is userImpl and provides functionalities for last level of code flow. UserImpl class implements User interface. It contains a Map which contains all the names of portfolios that exist in a particular directory against the Portfolio object which is null right now and gets keep on filling as portfolios are retrieved.

View Design

* There is only one class in View, i.e. ViewPrint which contains the functions, all of them just print based on user’s input.
* All the functions in view are static as there is no need of creation of View Object and functions can be called directly on class.