UNIVERSITY OF MARYLAND
COLLEGE PARK

ENSE 623
VALIDATION AND VERIFICATION

DMS-COP

Decision Making System for Construction of Optimal Portfolio

PROFESSOR

MARK A. AUSTIN

STUDENTS

ALI PILEHVAR

SUMAN SAMPATH

Tuesday, December 12, 2006

Contents

6-System validation and verification

1-Introduction

1-1 Problem statement

In today's financial marketplace, a well-maintained portfolio is vital to any investor's success. As an individual investor, you need to know how to determine an asset allocation which best conforms to your personal investment goals and strategies. In other words, your portfolio should meet your future needs for capital and give you peace of mind. Investors can construct portfolios aligned to their goals, their risk aversion and investment strategies by following a systematic approach.

The most important job for each investor to make his/her desired portfolio is asset allocation.
Asset allocation is an investment portfolio technique that aims to balance risk and create diversification by dividing assets among major categories such as cash, bonds, stocks, real estate and derivatives. Each asset class has different levels of return and risk, so each will behave differently over time. The consensus among most financial professionals is that asset allocation is one of the most important decisions that investors make. In other words, your selection of stocks or bonds is secondary to the way you allocate your assets to high and low-risk stocks.
You should keep in your mind that there is simple rule in the real financial world:” If you are interested to have a portfolio with the higher average return you should be ready for higher risk tolerance. You can see this simple rule in the figure1.

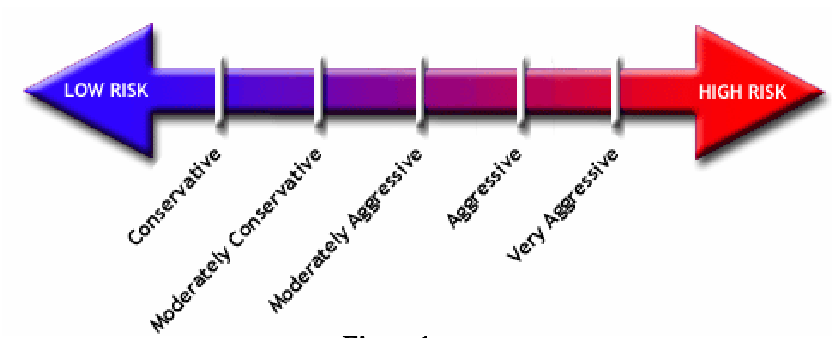


Figure1

1-2 Ideal solution

The scope of this project is to create a simple model-based program which can assist an individual investor to firstly construct an optimum financial portfolio to make more capital gain based on set of criteria and rules and secondly give his/her the tool to decide how to manage portfolio from the perspective of selling, buying and keeping portfolio’s securities or getting out of the market according to the risk-return characteristics that best suit one’s preference and financial objectives.
This system will consist of 3 main processes:

--The first one is the market analysis that can help us to make daily decision in regard to buy, sell or hold each asset by the end of the business day.
--The second one is the optimization problem that will construct the optimum portfolio from the point of Maximizing Return While minimizing the risk or standard variation of the portfolio.
The main goal of optimization process which is in other words some kind of the allocation of assets among various asset classes is to maximize return for your chosen level of risk, or stated another way, to minimize risk given a certain expected level of return.

--The third one is to determine the best number of shares which we should buy or sell for each stock based one the optimum obtained weights for each stock and also the result of the market analysis daily decision to achieve the best portfolio with the maximum return and with the minimum risk (volatility).
By combining these 3 processes we can have the optimum portfolio with the best possible asset allocation by the end of each business day in order to have the best portfolio by ending the investor given period.

This project follows the development of this decision-making tool included 2 main processes, from the generation and mapping of requirements, to the UML diagramming the sequences of events, to the generation of the program code in Matlab, Visual basic and Microsoft Excel, and finally to the verification and validation of the decision support tool.

1-3 High level objectives of the project

1-Systematicly determining and analyzing the market trend of a security based on historical data and some existing techniques.(Market analysis decision)


Ticker-T

Symbol	Last	Net	Bid	Ask	BSxAS
SUNW	5.292	0.402	5.29	5.30	142x203
INTC	19.23	1.48	19.23	19.24	12x84
ORCL	9.94	0.39	9.94	9.95	13x355
AMAT	19.56	1.57	19.56	19.57	61x20
QCOM	28.05	1.67	28.04	28.05	3x5
DELL	25.72	1.04	25.71	25.72	37x1
JDSU	2.877	0.297	2.86	2.88	59x553
VRTS	21.69	1.95	21.68	21.69	33x1
JNPR	7.06	0.36	7.06	7.07	24x32
KLAC	45.24	2.77	45.22	45.23	8x14
NVDA	18.69	1.24	18.71	18.72	9x1
BRCD	17.38	1.19	17.38	17.40	22x11
MXIM	40.05	2.28	40.04	40.06	26x1
XLNX	23.13	1.67	23.06	23.13	88x1
MCHP	27.35	1.92	27.34	27.38	20x3

Figure2

2-Constructing an optimum portfolio of different kinds of investments which can, on average, yield higher returns and pose a lower risk than any individual investment found within the portfolio according to scenarios presented by the market analysis.(optimum portfolio allocation)

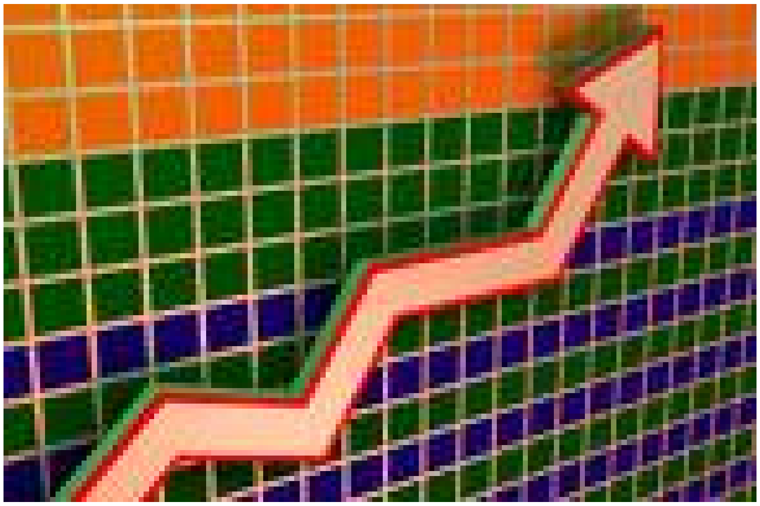


Figure3

3-Make use of the benefits of diversification techniques to have a portfolio whose securities are even negatively correlated to smooth out unsystematic risk events. The more you have stocks in you portfolio the more you can lower your portfolio variance and risk.

3-to achieve a portfolio with the minimum risk for the given expected return and other initial information.

1-4 Assumptions

The scope of this system will assume the following conditions.

1-Here among all of the securities can be used to construct a portfolio we preferably choose among the stocks due to their further uncertain and stochastic behaviors.

2-We will assume that each portfolio consists of just 4 stocks.

3- The maximum length of the market analysis and balancing of the portfolio for this system is 1 year since we assume that the market risk, interest rate investor risk-tolerance is constant for all these 12 month timeframe.

4- We assume that this system do the optimization just once by beginning of the investment period to calculate the optimum weight of these stocks in the portfolio. By passing every day we do market analysis independently for each stock to see whether we should buy, sell or hold stocks, then by means of new closed price we can say how many share of each stock we should buy or sell.

5- The daily market just give us the latest price information for chosen stocks needed for calculating the new number of share for each stock by passing one day.

6- We assume we will make profit by long term sale not the short sale.

7-We assume that we can buy and sell stocks in the beginning of the next business day with the today's closing price, but in the real stock exchange world you are not guaranteed to get this price if you buy the stock the next day because the price is constantly changing (even after the exchange is closed for the day).

8-In our system we assume that the portfolio rebalancing strategy is daily basis according to the market analysis process. This policy is consisting of changing the shares of these 4 stocks by buying, selling and holding scenarios in order to construct a new portfolio with The same weights of each stock resulting in achieving the least portfolio's risk (volatility) at the end of the time frame if we try to stick to those weight by appropriate number of sell and buy of each stock.
9- We assume that in our system we ignore the effects of any tax and other fees such as trading costs, operating expenses, commission costs, etc in our daily selling or buying the stocks. We also assume that the interest rate is unchanged during 1 year.

10- Our chosen stocks can have non-zero covariance with each other, in other words the stocks can have dependency with each other and by this way we can take advantage of reduction the risk (total standard deviation) of the portfolio by means of the diversification.

11-From now on, we name the *Decision making system for constructing the optimum portfolio* DMS-COP.

12- If the result of the market analysis for a stock is to hold that stock, we assume the new closing price of that stock is the same as the previous day since they are so close.

13- The daily market actor in use case diagram and system boundary is considered as an online resource that we obtain the daily closing price of each stock from it.

1-5 Terminology

Portfolio: The group of assets - such as stocks, bonds and mutual funds - held by an investor.

Stocks: common stocks also known as equity securities, represent ownership shares in a corporation. Shareholders have voting rights and may receive dividends.

Short sale:
A short sale allows investors to profit from a decline in a security's price.

Long term sale:
A long term sale allows investors to profit from an increase in a security's price.

Rate of return:
A key measure of investors' success is the rate at which their portfolios have grown during the investment period.

Rebalancing the portfolio:
The process of realigning the weightings of one portfolio.

Risk (Uncertainty) of a stock:
Each stock has its own standard deviation from the mean, which MPT [1]calls "risk".

Weight of stocks:
The total value of each stock in the portfolio which is simply the multiplication of number of each stock's shares by their prices divided by the total value of portfolio. It is a number between 0 and 1 for each stock.

Volatility:
A statistical measure of the dispersion of returns for a given security or market index. Volatility can either be measured by using the standard deviation or variance between returns from that same security or market index. Commonly, the higher the volatility, the riskier the security.

Total Portfolio Volatility:
It is the equivalent variance or standard variation of all stocks in a portfolio. It can be minimized if Stocks return in a portfolio is not correlated to each other.

Standard deviation:
A measure of the dispersion of a set of data from its mean. The more spread apart the data is, the higher the deviation. In finance, standard deviation is applied to the annual rate of return of an investment to measure the investment's volatility (risk).

Volatile stocks:
Those stocks with high standard deviation.

Risk tolerance:
The degree of uncertainty that an investor can handle in regards to a negative change in the value of their portfolio.

Diversification: A risk-management technique that mixes a wide variety of investments within a portfolio. The rationale behind this technique contends that a portfolio of different kinds of investments will, on average, yield higher returns and pose a lower <<lower volatility>> than any individual investment found within the portfolio. Positive correlation among securities in a portfolio can reduce the benefits of portfolio diversification and increase the total portfolio volatility.
For a well-diversified portfolio, the risk - or average deviation from the mean - of each stock contributes little to portfolio risk. Instead, it is the difference - or covariance - between individual stocks' levels of risk that determines overall portfolio risk. As a result, investors benefit from holding diversified portfolios instead of individual stocks.

[1] MPT is (modern portfolio theory)

Unsystematic Risk - Also known as "specific risk", this risk is specific to individual stocks and can be diversified away as you increase the number of stocks in your portfolio (see Figure 1). It represents the component of a stock's return that is not correlated with general market moves.

Efficient frontier:
A line created from the risk-reward graph, comprised of optimal portfolios.

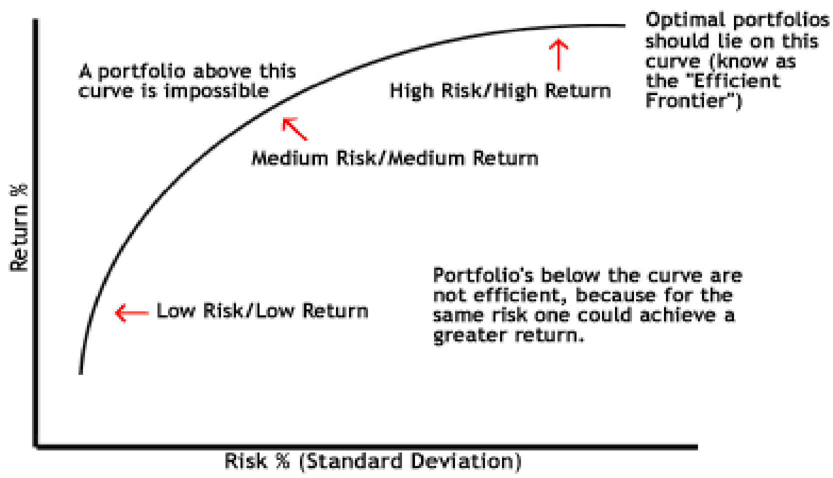


Figure4

For every level of return, there is one portfolio that offers the lowest possible risk, and for every level of risk, there is a portfolio that offers the highest return. These combinations can be plotted on a graph, and the resulting line is the efficient frontier
The optimal portfolios plotted along the curve have the highest expected return possible for the given amount of risk.

ADX – Average Directional Index: An indicator used to determine the strength of the trend. It is measured on a scale between zero and 100.

DMI - Directional Movement Index: Component of ADX indicator. Helps in buy/sell determination.

Trending – When a particular asset's direction continues to move in one direction into the future.

MA – Moving Average: An indicator showing the average value of a stock price over a set period.

Oscillating – Erratic moving behaviour of a particular asset.

%K line – component of stochastic indicator: moving average of 5 to 10 day period

%D line – 3-period moving average of %K line.

Bollinger Band – A band plotted two standard deviation away from a simple moving average. When markets become more volatile, the bands widen and vice versa.

Ticker – Name of the stock

Closing Price – The price at the end of the day.

1-6 System framework and Boundary

Our decision making system is composed of interaction between actors as depicted in the diagram bellow and the outside financial environment.

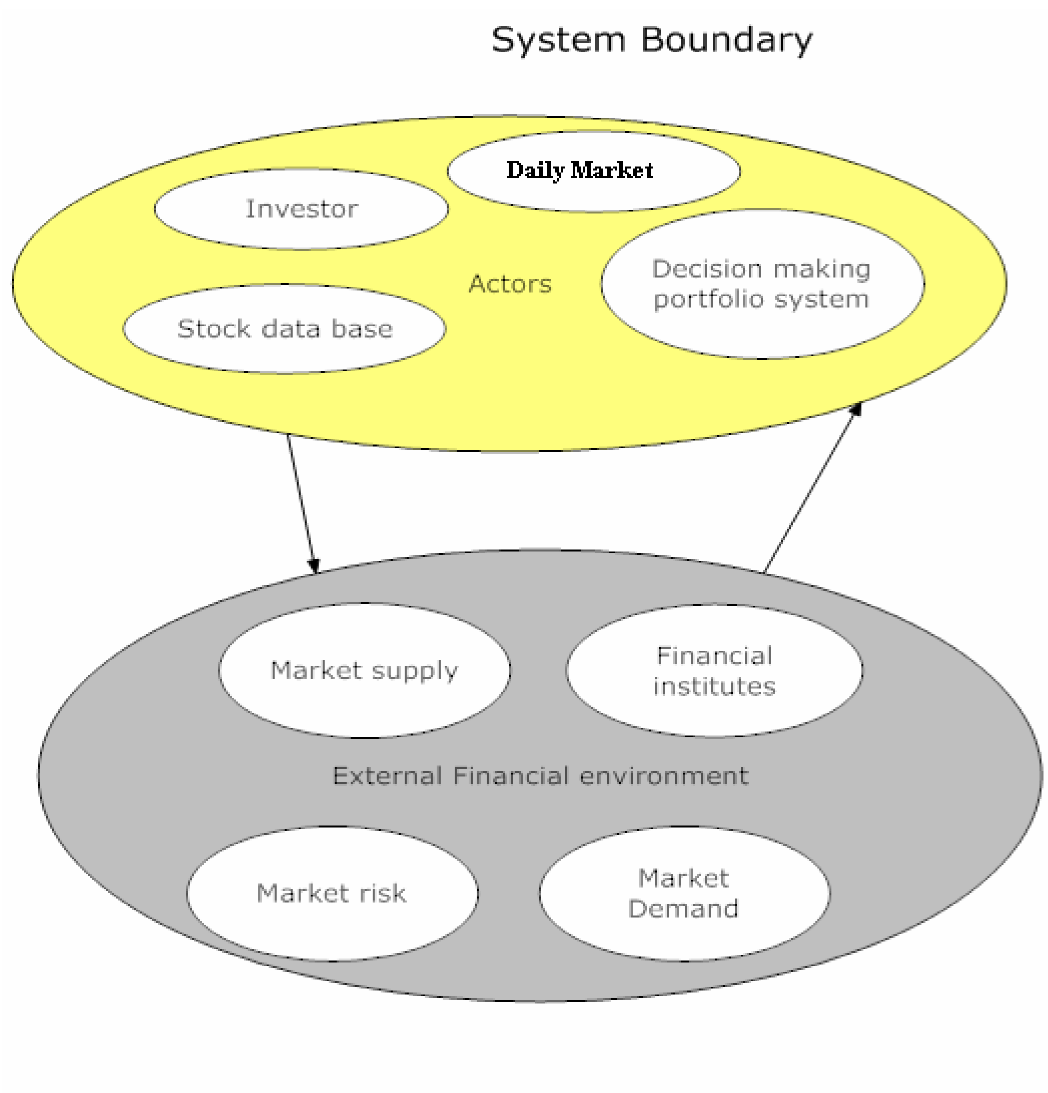


Figure5

2-Goals, scenarios and use cases

2-1 Goals and scenarios

1-Goal1: The system must be user friendly.

Scenario1-1-The system must be easy to use
Scenario1-2 The investor must easily put their initial information to the system.
Scenario1-3 the system should demonstrate and report the results of each day as simple as possible the investor.

2- Goal2: The system must be reliable.

Scenario2-1-The system should obtain the historical for chosen tickers from stock data base from the sincere resources.
Scenario2-2The system should obtain the daily close prices for each stock from the daily market data bases efficiently.
Scenario2-3The system must be economic

3- Goal3: The system should determine the optimized allocation for the set of the given stocks.

Scenario3-1The system should obtain the scenario satisfied the minimum the expected portfolio return.
Scenario3-2The system should obtain the least possible risk for then given initial portfolio.
Scenario3-3The system should calculate the new number of shares from buy, sell and hold decision to guarantee achieving the minimum portfolio's risk and meeting the expected average return along with keep going to stratify the optimum stocks weights.
Scenario3-4The system' obtained portfolio at any time should meet the expected given return as well as the obtained optimum stocks weight in the portfolio allocation.

4- Goal4: The system should analyze the market of each given stock by a known technique to find out 3 potential solutions (Buy/Sell/Hold) for each stock.

Scenario4-1The system decides on how the average directional movement index is going.

5-Goal5: The results of the market analysis should be correspondent to the results of the optimization and also the outcomes from determining the number of each stock's shares.

Scenario5-1 After knowing about the buy, sell or hold decision for each stock, the system should calculate the number of stocks; the investor should sell or buy.

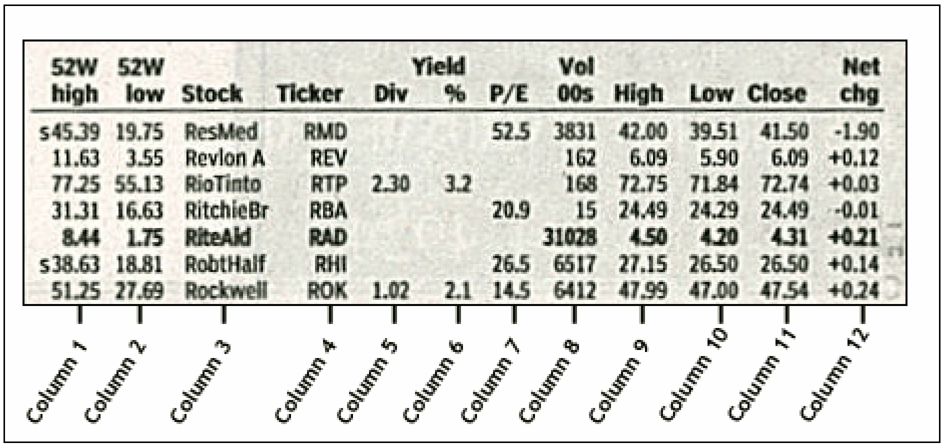
2-2 Use case analysis

2-2-1 Identify the actors

Actors:

1-**Investor**: person who desire to make decision on how to construct the best portfolio.

2-**Stock database**: Is the pool of the historical data for stocks for the last several years. You can see the sample of the data table structure of these stocks in the following graphs. But you should know that we make use of the parts of this information for each stock in our project, because some of these items are further than the scope of this project.



52W high	52W low	Stock	Ticker	Div	Yield %	P/E	Vol 00s	High	Low	Close	Net chg
s45.39	19.75	ResMed	RMD			52.5	3831	42.00	39.51	41.50	-1.90
11.63	3.55	Revlon A	REV				162	6.09	5.90	6.09	+0.12
77.25	55.13	RioTinto	RTP	2.30	3.2		168	72.75	71.84	72.74	+0.03
31.31	16.63	RitchieBr	RBA			20.9	15	24.49	24.29	24.49	-0.01
8.44	**1.75**	**RiteAid**	**RAD**				**31028**	**4.50**	**4.20**	**4.31**	**+0.21**
s38.63	18.81	RobtHalf	RHI			26.5	6517	27.15	26.50	26.50	+0.14
51.25	27.69	Rockwell	ROK	1.02	2.1	14.5	6412	47.99	47.00	47.54	+0.24

Figure6

The stocks data base also consist of the average percentage return of the stocks for previous years along with their standard deviation.

You can see more explanation for the data used in our project in the appendix.

3-The daily market: it is the online resource of information to achieve the latest price of the given stocks needed in the calculating the number of buy and sell for each stock in the process of the calculation the best number of buy or sell when you know the daily market analysis decision.

2-2-2 Use case diagram

A use case describes system functionality in terms of users inputting information to or receiving results from the system. Although use cases are neither requirements nor functional specification, they demonstrate the requirements, objects, and object interactions briefly. Use cases are textual descriptions of the interactions between external actors and the system itself.

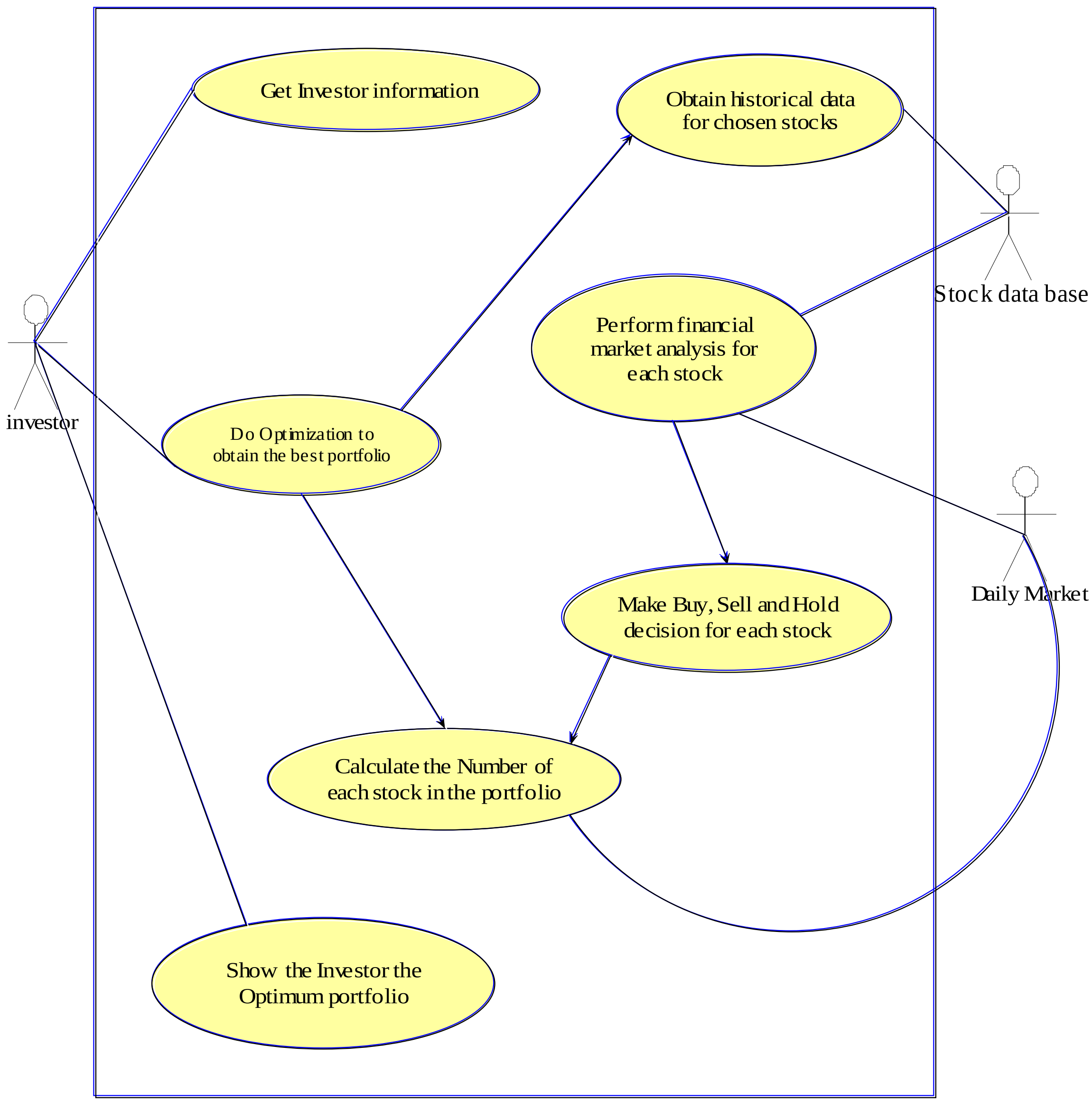


Figure 7

2-2-3 Use case description:

1. **Use Case: Get investor information**

Description: An investor gives the system his/her asked information.
Primary Actors: Investor
Pre-conditions: An investor has already known that what is his/her initial amount of investment, minimum expected return, upper bound and lower bound of his/her favorite stocks' weights and names of stocks he/she desires to have in the portfolio.
Flow of Events: 1. The DMS-COP[1] asks for entering the initial amount of the investment, stock names along with their upper and lower bound of the preferred weights of each of these stocks and minimum expected return. 2. The Investor enters his/her initial amount of investment to the system. 3. The investor enters the Ticker of the chosen stocks. 4. The investor enters the minimum expected return for the timeframe. 5- The investor enters the lower and upper bound preferred weights for each stock. **Post Condition:** Investor successfully enters his/her information to the system.

2. **Use Case: Do Optimization to obtain the best portfolio**

Description: It is the process of doing the optimization at the beginning of the running the system for the first time to come up with the initial optimum portfolio. To objective of this use case is initially determining the optimum stocks' weights and finally calculating the best number of buy or sell for each stock.
Primary Actors: Investor, Stocks data base and daily market.
Pre-conditions: The system has already known investor' initial amount of investment, minimum expected return ,name of stocks (tickers) with the desirable max and min weights for those tickers in the portfolio.
Flow of Events: 1.The DMS-COP initiates **Obtain historical data for chosen stocks** <<includes>> acquiring average annual return, covariance between stocks, and average standard deviation for chosen stocks from the stock data base.

[1] The Decision making system for constructing the an optimum portfolio

2. The DMS-COP solves the optimization by replacing the entered information in the parameters of the optimization model to achieve the best possible weights resulting in the least portfolio risk for the given expected return. 3.The DMS-COP initiates **Calculate the Number of each stock in the portfolio** <<Includes>>. 3.1 The DMS-COP get the latest closing price for each stock from the daily market. 3.2 The DMS-COP obtained the results of the daily market analysis for each stock whether buy, sell or hold. 3.3. The DMS-COP calculates the number of each stock in the portfolio to meet the calculated weight for each stock 3.3 The DMS-COP record the new obtained number of share of each stock as the best optimum portfolio at that time.
Post Condition:
The system will successfully construct the best possible portfolio possessing the least risk volatility satisfying the given expected return.

3. Use Case: Show the Investor the Optimum portfolio

Description: It is the use case showing how the system will show the investor the optimum portfolio.
Primary Actors: Investor
Pre-conditions: The system has found out the new optimum portfolio at the end of each day by knowing the latest closing price.
Flow of Events: 1. The DMS-COP will show the investor that by the end of the day what is the optimum number of shares of stocks to achieve the least portfolio by satisfying the buy , sell and hold decision carried out in the market analysis.

4. Use Case: Perform financial market analysis for each stock

Description: The system gives Buy/Sell/Hold decision outputs based on the market analysis performed by the system and the inputs of securities given by the user. When the user enters certain securities like MSFT, YHOO, IBM etc, the system connects to the data base already gathered from financial market like Yahoo Financials or Bloomberg and fetches the historical data. The system then does a series of calculations – Moving Average, Bollinger, Stochastic Process, and ADX – to conclude if a particular security should be a buy or sell or hold.
Primary Actors:

Investor, DMS-COP, Stock Database, and daily market.
Pre-conditions: Because this is a physical system that also gathers data prior to doing any calculations there are a number of inputs the system needs before it can perform market analysis: 1. The system must have the exact data base at all times or at least at the time of gathering data. 2. From the daily market the system must collect data on a daily basis in regard to closing price of the stocks.
Flow of Events: <u>The Investor</u> 1. The investor opens the DMS-COP Application. 2. The investor opens the database where the historical data of all the stocks have been retrieved. 3. The investor enters the Stocks he would like to invest in – MSFT, YHOO, IBM. 4. Investor clicks on “Run Model” to run the system (DMS-COP) <u>The System</u> 1. Retrieves the historical data of the appropriate stocks from the Database. 2. Performs ADX calculation dictated by the ADX model. 3. If the system determines that a particular stock is Trending, perform appropriate calculations to determine whether to buy/sell/hold a particular stock. 4. If the system determines that a particular stock is Oscillating, the system performs appropriate calculations to determine whether to buy/sell/hold a particular stock.
Post Condition: The system outputs the buy/sell/hold decisions on each stock. The user uses this information to further optimize his decision making by using the optimization procedure.

3- Generation of the requirements

After generating the textual use cases and the scenarios related to those use cases and goals, requirements can be evaluated. Requirements are derived in a top down format from the goals and scenarios; it is crucial to find out the connection between requirements and use cases and maintain integrated traceability between them in order to have a strong basis for system behavior and structure modeling in the next sections.

3-1General requirements

A- Investor requirement

1- The investor should know what his /her initial investment is, minimum expected return, upper and lower bound desired weights of each stock, and the list of investor's favorite stocks.
2- The investor should understand exactly the report feature of the result.

B- System requirement

1-The system should have user friendly feature and format. (User-friendly requirement).

2- The system should obtain the optimized weights of each stocks (best allocations) to satisfy the investor's requirement.(Optimization requirement).

3- The system should analyze the market of each given stock by a known technique based on the valid historical data in order to be in the same direction of the optimization requirement (market analysis requirement).

4-The system should calculate the optimum number of buy or sell for each stocks in order to satisfy the optimum stock's weights at any time.

5-The system should obtain the needed historical data for chosen stocks efficiently.

6- The system should have access to the daily closing price of each stock in the portfolio

3-2 Requirement traceability

Tracing of requirements back to originating use cases is as follows.

3-2-1 the follow-down requirement requirements from use cases

Use Case	No. Req.	Requirements Description
Get investor information (1)	A1	The investor must know his/her initial amount of investment , the name of his/her desired ticker; the minimum and maximum preferred stock' s weights in the portfolio and finally knows the minimum expected return rate from that portfolio .
	B1	The system should be designed and manipulated in a user-friendly format such as Microsoft excel which most people knows a little a bit about it.

Table1

Use Case	No. Req.	Requirements Description
Do Optimization to obtain the best portfolio (2)	B1	The system should be designed and manipulated in a user-friendly format such as Microsoft excel which most people knows a little a bit about it
	B2	The system should construct the minimum optimization model to minimize the risk or overall standard deviation of the portfolio for a given expected return, upper an lower bound of each stock's weights.
	B3	The system should make us of the decision results from market analysis in order to calculate the best number of the buy, sell of each stock.
	B4	The system should calculate the best the number of buy or sell shares for each stock after obtain the latest closing price of each stock and the results from market analysis.
	B5	The system should have efficient access to the historical data needed to satisfy the B2. The system should have the historical data in excel.
	B6	The system should have the efficient access to the daily market to obtain the latest closing price needed in B4.

Table2

.

Use Case	No. Req.	Requirements Description
Show the Investor the Optimum portfolio (3)	B1	The system should be designed and manipulated in a user-friendly format such as Microsoft excel which most people knows a little a bit about it
	A2	The investor should e informed about content of the report in order to understand the results perfectly for implementing the daily decision.

Table 3

Use Case	No. Req.	Requirements Description
Perform financial market analysis for each stock (4)	B1	The system should be designed and manipulated in a user-friendly format such as Microsoft excel which most people knows a little a bit about it
	B3	The system should analyze the market of each given stock by a known technique based on the valid historical data.
	B5	The system should have efficient access to the historical data needed to satisfy the B3. The system should have the historical data in excel.

Table 4

3-2-2 Traceability of the requirement to the use cases

Requirement #	Requirement Description	Use case #
A1	The investor must know his/her initial amount of investment , the name of his/her desired ticker; the minimum and maximum preferred stock' s weights in the portfolio and finally knows the minimum expected return rate from that portfolio .	1
A2	The investor should be informed about content of the report in order to understand the results perfectly for implementing the daily decision.	3
B1	The system should be designed and manipulated in a user-friendly format such as Microsoft excel which most people knows a little a bit about it.	1,2,3,4
B2	The system should construct the minimum optimization model to minimize the risk or overall standard deviation of the portfolio for a given expected return, upper and lower bound of each stock's weights.	2
B3	The system should analyze the market of each given stock by a known technique based on the valid historical data.	2,4
B4	The system should calculate the best the number of buy or sell shares for each stock after obtain the latest closing price of each stock and the results from market analysis.	2
B5	The system should have efficient access to the historical data needed to satisfy the B2. The system should have the historical data in excel.	2
B6	The system should have the efficient access to the daily market to obtain the latest closing price needed in B4.	2

Table5

4-System modeling and behavior

4-1 High level behavior

System behavior covers identification of key functions and ordering and decomposition of these functions. It is the ordering of these functions coupled with the system logic that generate the system behavior.

4-1-1 System activity diagram

Since the most important processes in our system is optimization and market analysis, we just show the activity diagrams for these 2 and their sub-procedures.

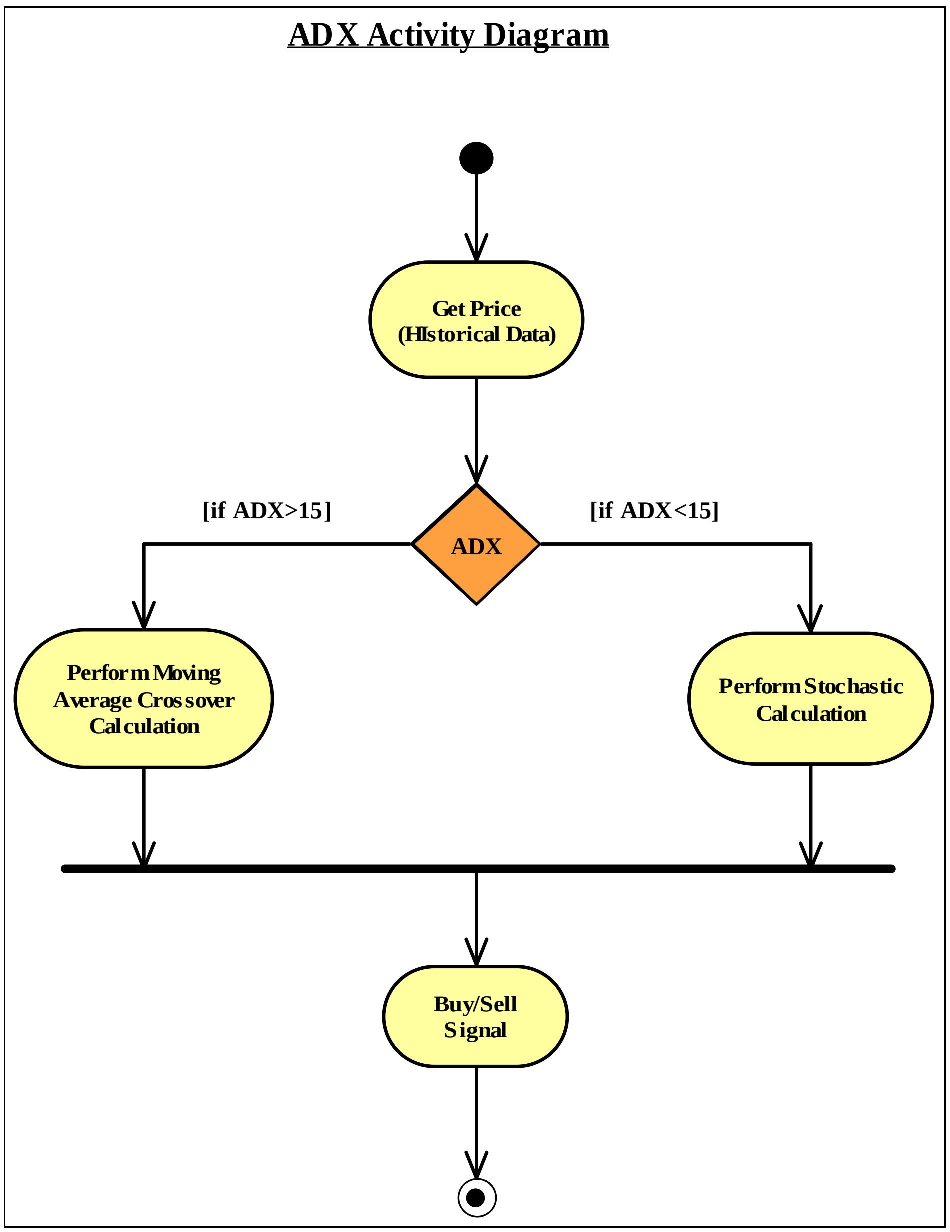


Figure8

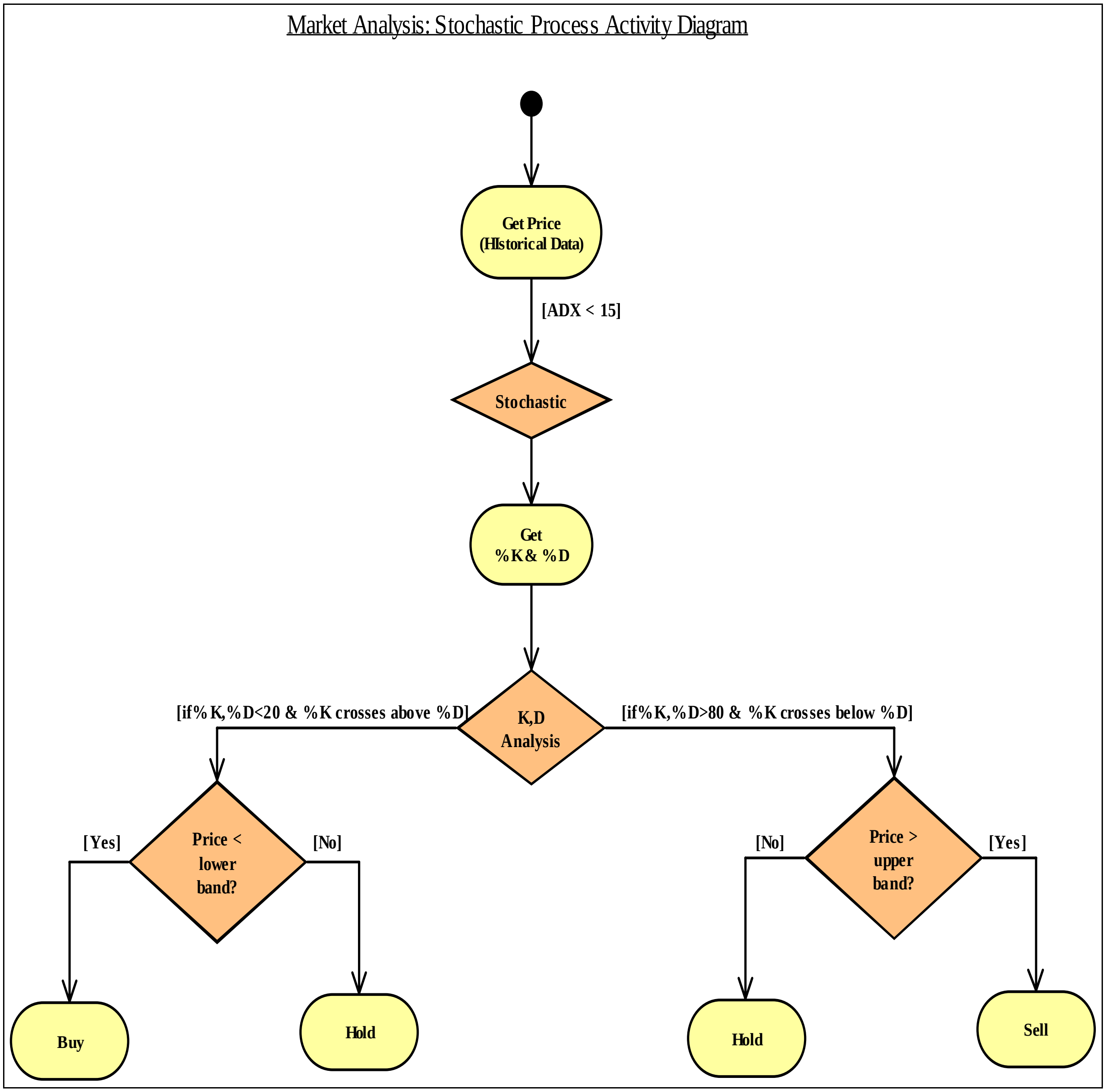

Market Analysis: Stochastic Process Activity Diagram
Get Price
(HIstorical Data)
[ADX < 15]
Stochastic
Get
%K& %D
[if%K,%D<20 & %K crosses above %D]
K,D
Analysis
[if%K,%D>80 & %K crosses below %D]
[Yes]
Price <
lower
band?
[No]
[No]
Price >
upper
band?
[Yes]
Buy
Hold
Hold
Sell

Figure9

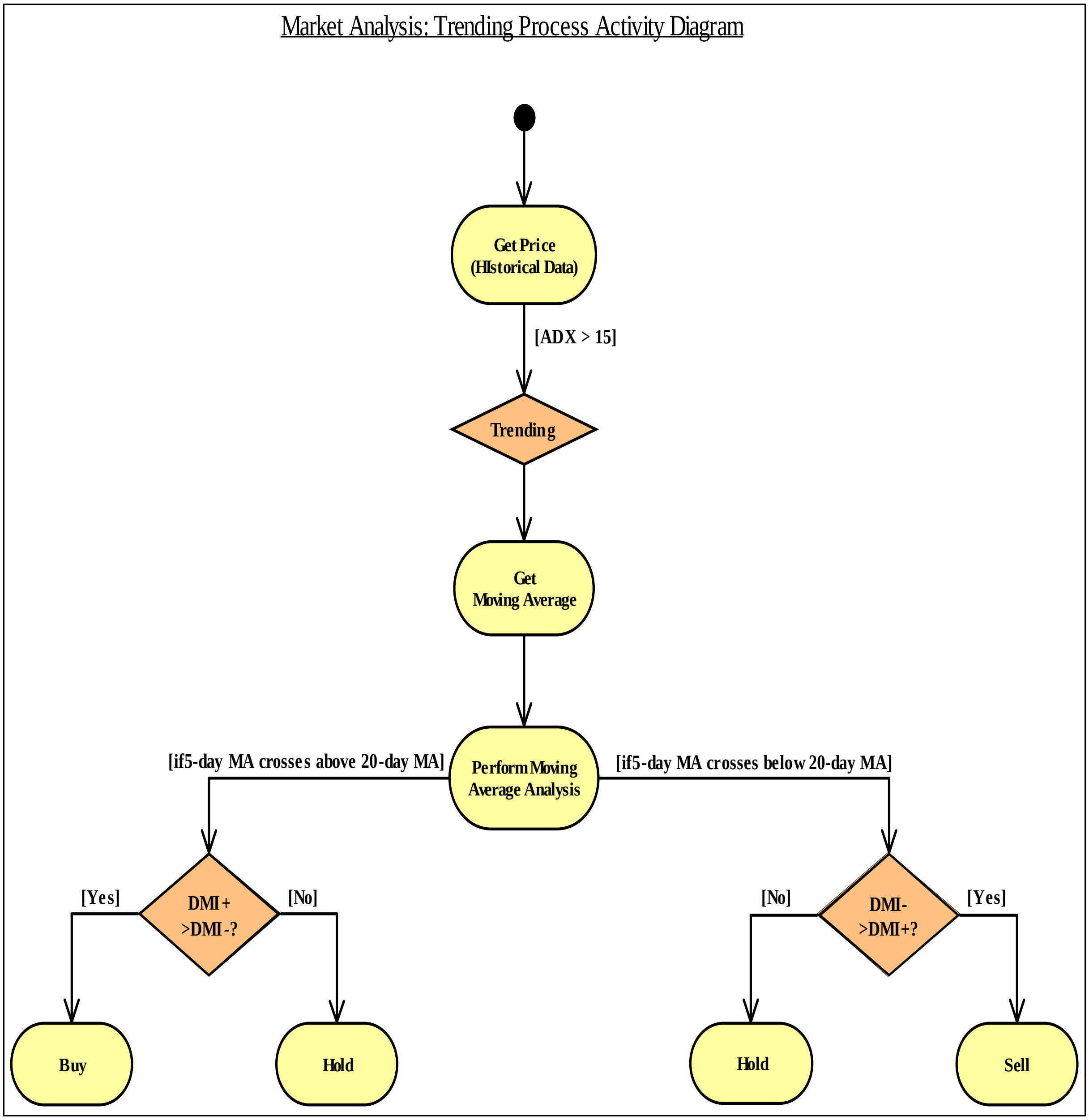

Market Analysis: Trending Process Activity Diagram
Get Price
(HIstorical Data)
[ADX > 15]
Trending
Get
Moving Average
Perform Moving
Average Analysis
[if5-day MA crosses above 20-day MA]
[if5-day MA crosses below 20-day MA]
[Yes]
DMI+
>DMI-?
[No]
[No]
DMI-
>DMI+?
[Yes]
Buy
Hold
Hold
Sell

Figure10

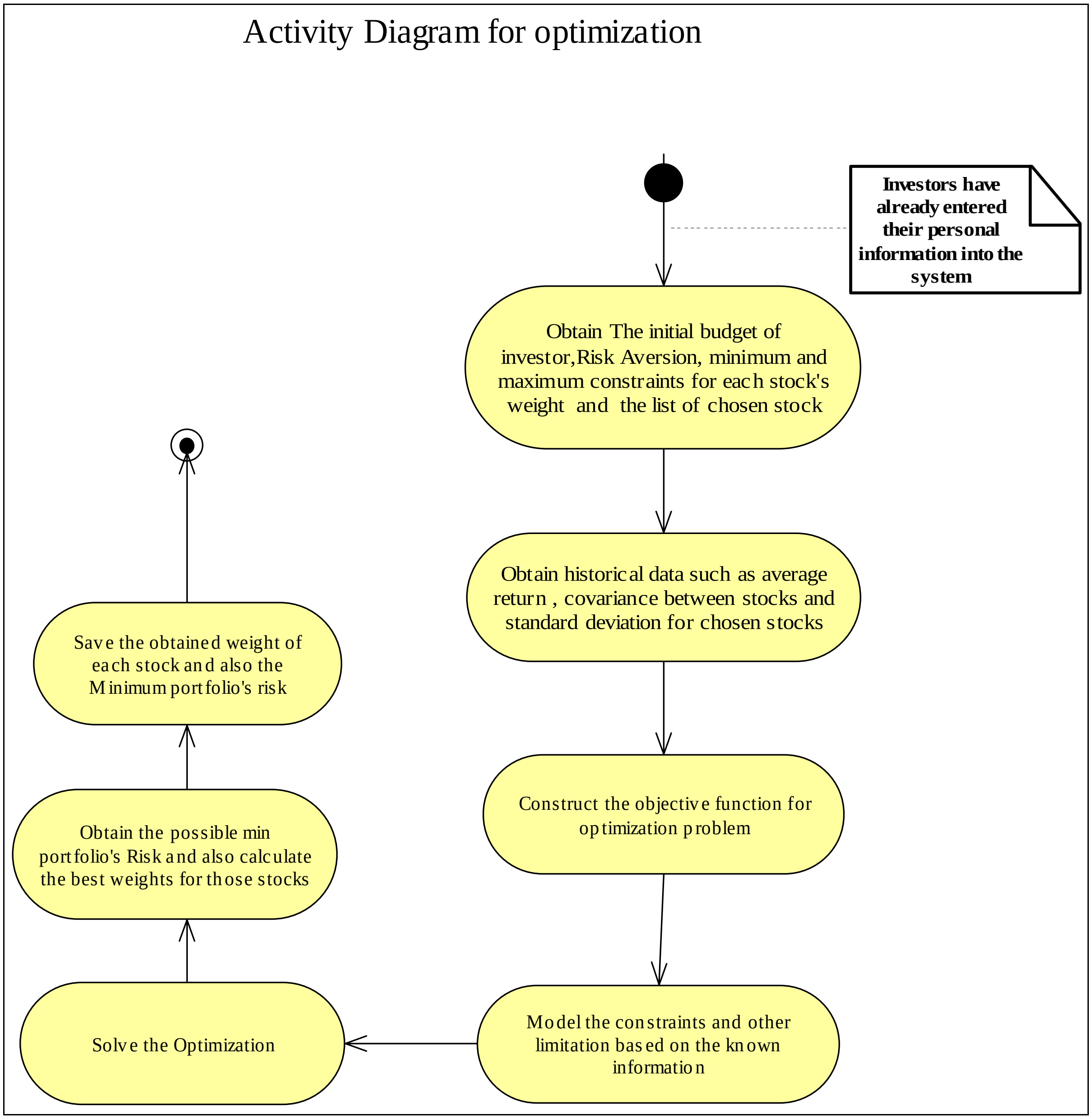
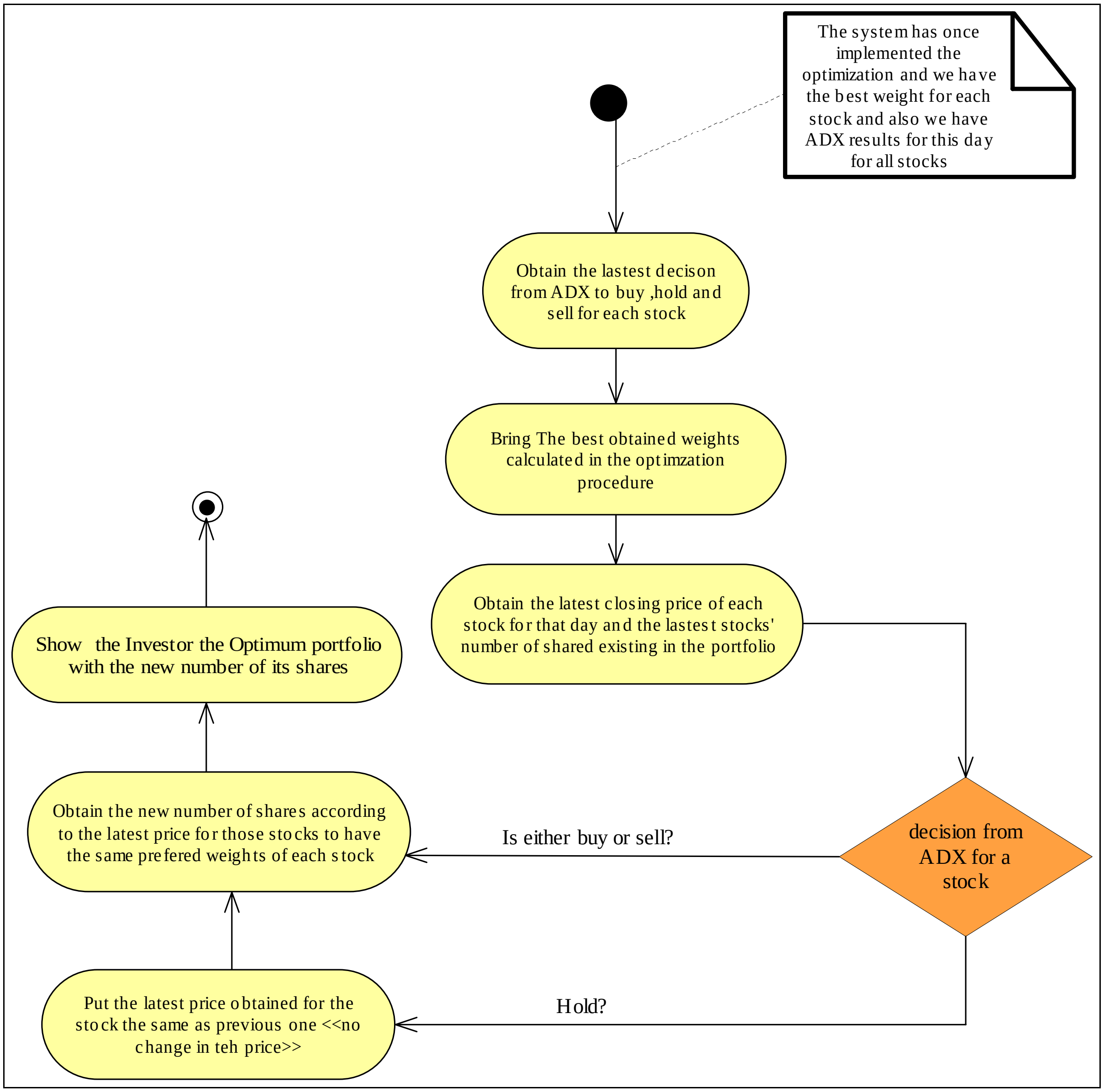

Activity Diagram for optimization
Investors have already entered their personal information into the system
Obtain The initial budget of investor,Risk Aversion, minimum and maximum constraints for each stock's weight and the list of chosen stock
Obtain historical data such as average return , covariance between stocks and standard deviation for chosen stocks
Construct the objective function for optimization problem
Model the constraints and other limitation based on the known information
Solve the Optimization
Obtain the possible min portfolio's Risk and also calculate the best weights for those stocks
Save the obtained weight of each stock and also the Minimum portfolio's risk

Figure11



The system has once implemented the optimization and we have the best weight for each stock and also we have ADX results for this day for all stocks
Obtain the lastest decison from ADX to buy ,hold and sell for each stock
Bring The best obtained weights calculated in the optimzation procedure
Obtain the latest closing price of each stock for that day and the lastest stocks' number of shared existing in the portfolio
decision from ADX for a stock
Is either buy or sell?
Hold?
Obtain the new number of shares according to the latest price for those stocks to have the same prefered weights of each stock
Put the latest price obtained for the stock the same as previous one <<no change in teh price>>
Show the Investor the Optimum portfolio with the new number of its shares

Figure12

4-1-2 System sequence diagram

Sequence diagrams provide a graphical representation for how a task is accomplished by passing a sequence of messages between objects of our system. Here we just show the sequence diagram for the optimization process.

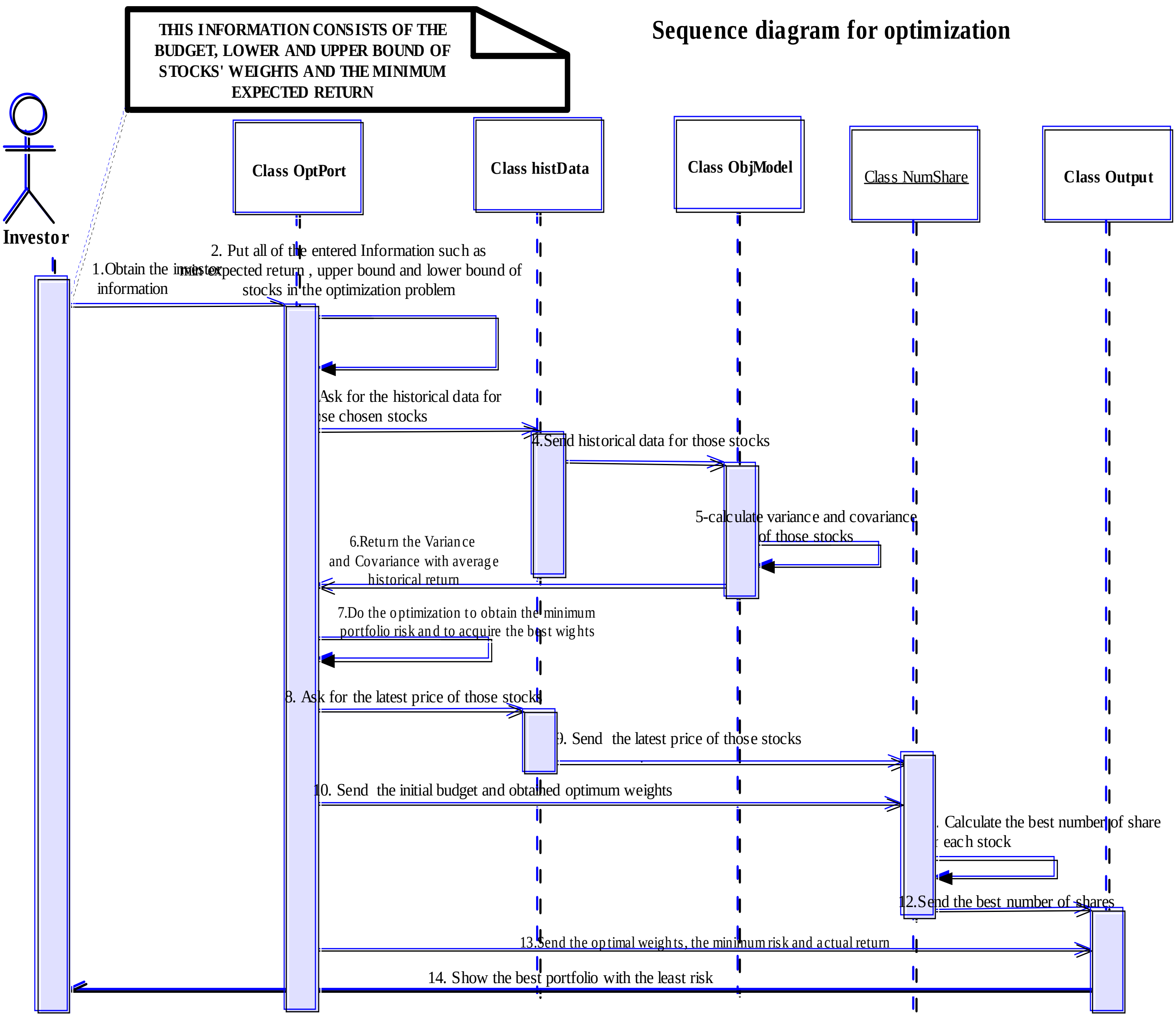


Figure13

4-2 High Level Structure

System structure covers decomposition of the system into hierarchies of principal subsystems, identification of required connectivity of components.

4-2-1 Class diagram

The diagrams below illustrate the classes and functions within the system. Since the most Important processes in our system are optimization and market analysis; we just show the class diagram of these 2.

It should be said again that the scope of this project is generating the structure of the software code based on behavioral modeling and analysis.

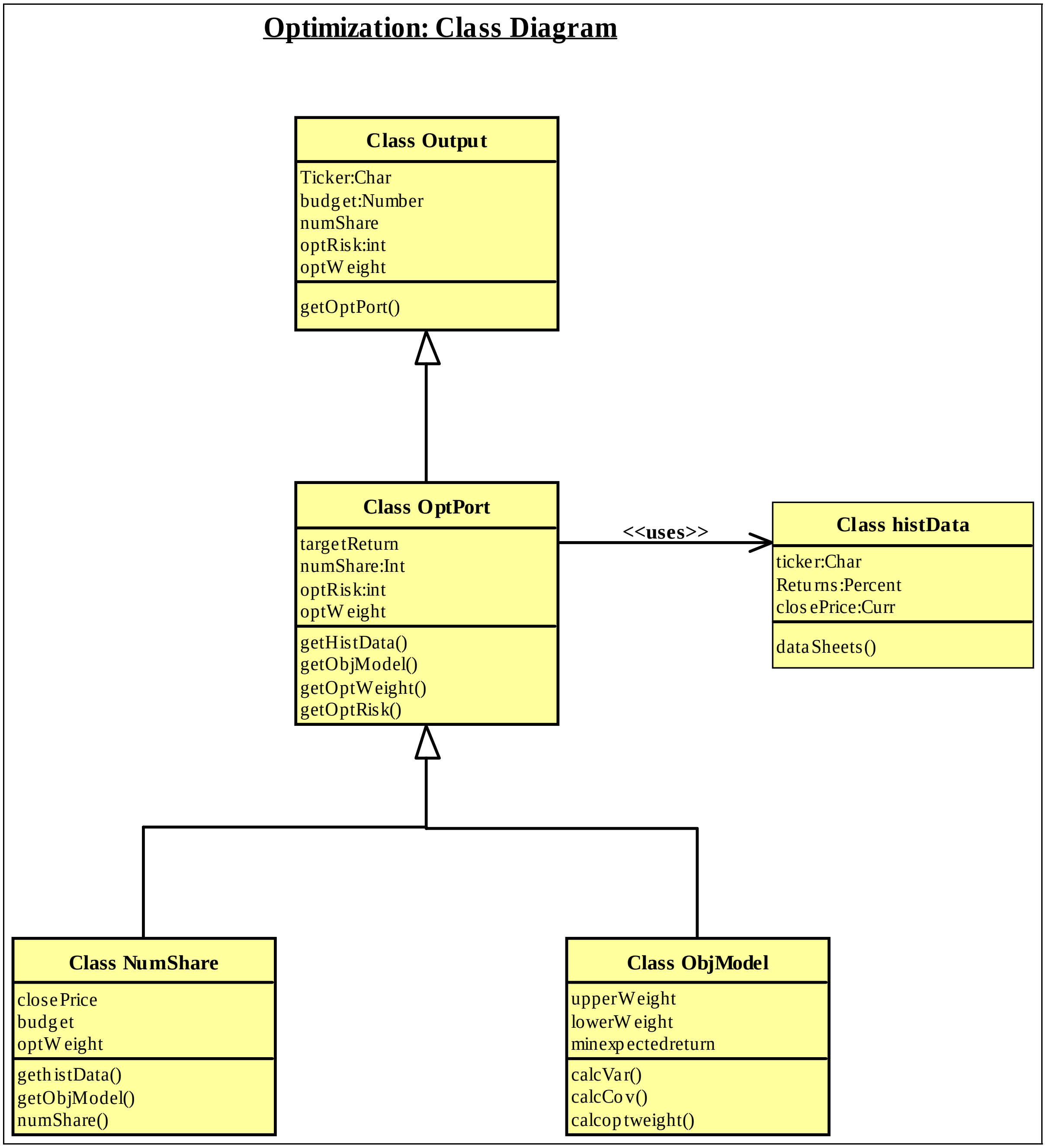


Figure14

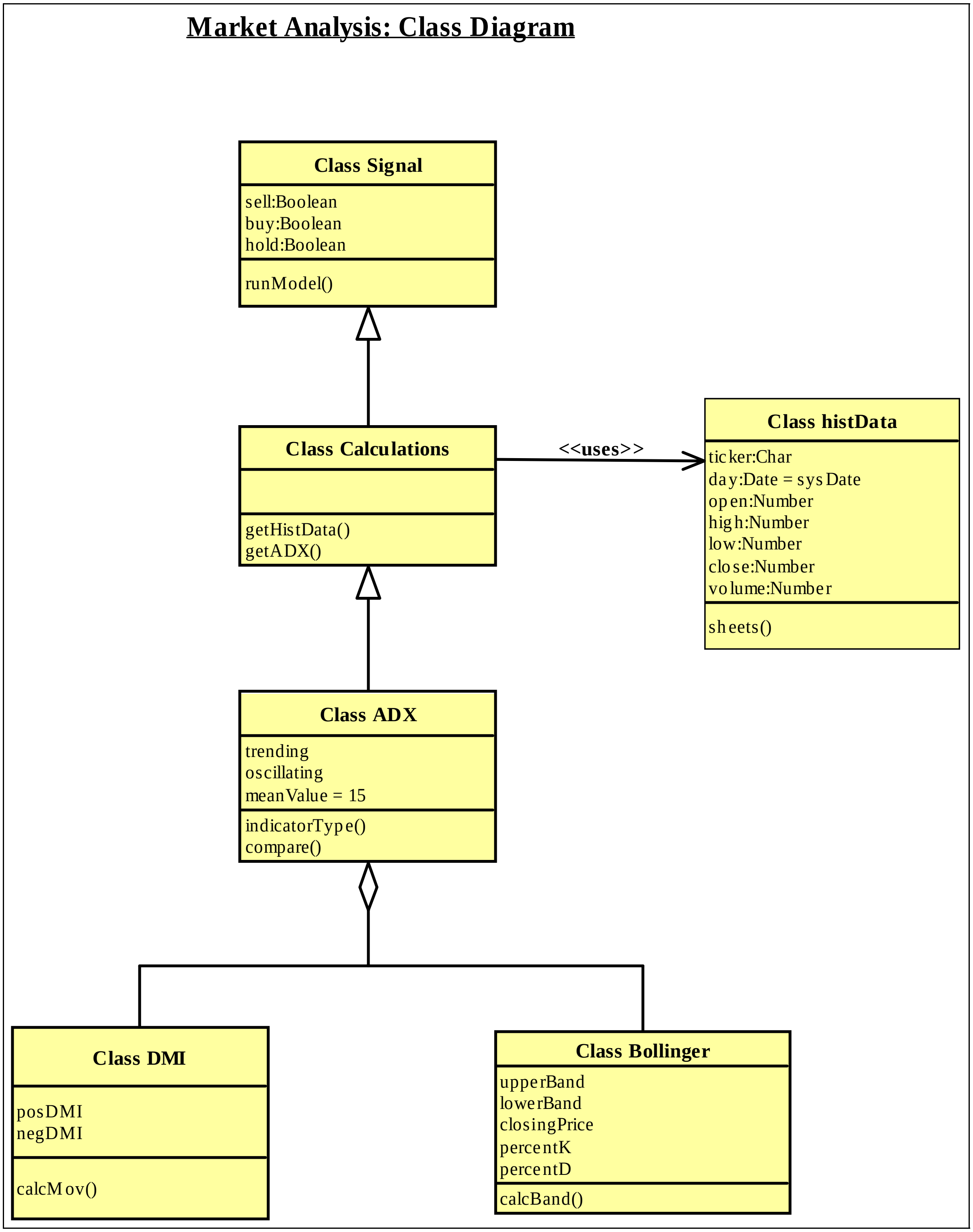

Market Analysis: Class Diagram
Class Signal
sell:Boolean
buy:Boolean
hold:Boolean
runModel()
Class Calculations
getHistData()
getADX()
<<uses>>
Class histData
ticker:Char
day:Date = sysDate
open:Number
high:Number
low:Number
close:Number
volume:Number
sheets()
Class ADX
trending
oscillating
meanValue = 15
indicatorType()
compare()
Class DMI
posDMI
negDMI
calcMov()
Class Bollinger
upperBand
lowerBand
closingPrice
percentK
percentD
calcBand()

Figure15

4-2-2 state chart

State charts show a behavioral view of the system. Here we just have the state chart for the market analysis process and its sub-procedures since we can define the states better and more meaningfully.

States of each stock after market analysis is buy, sell or hold.

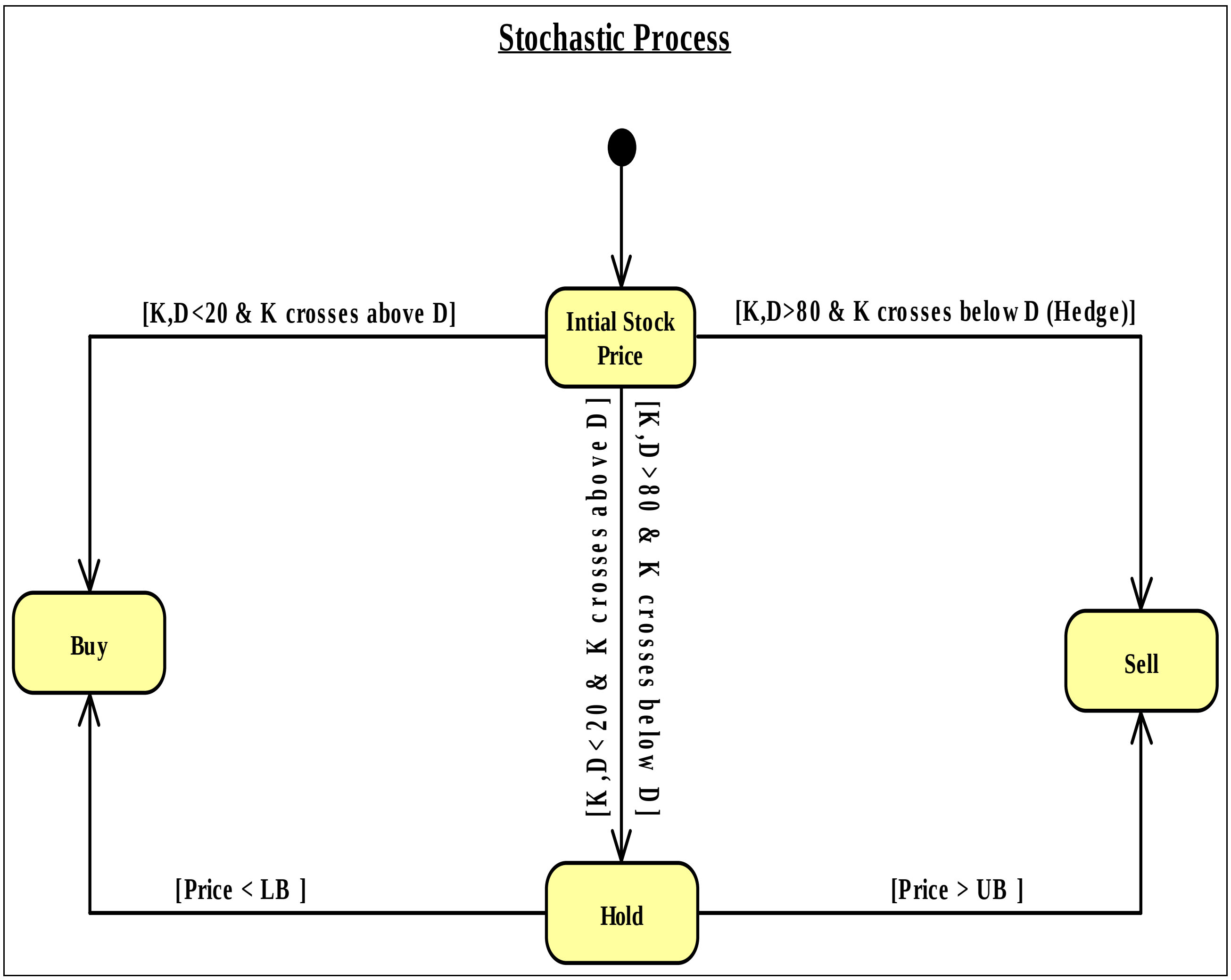


Figure16

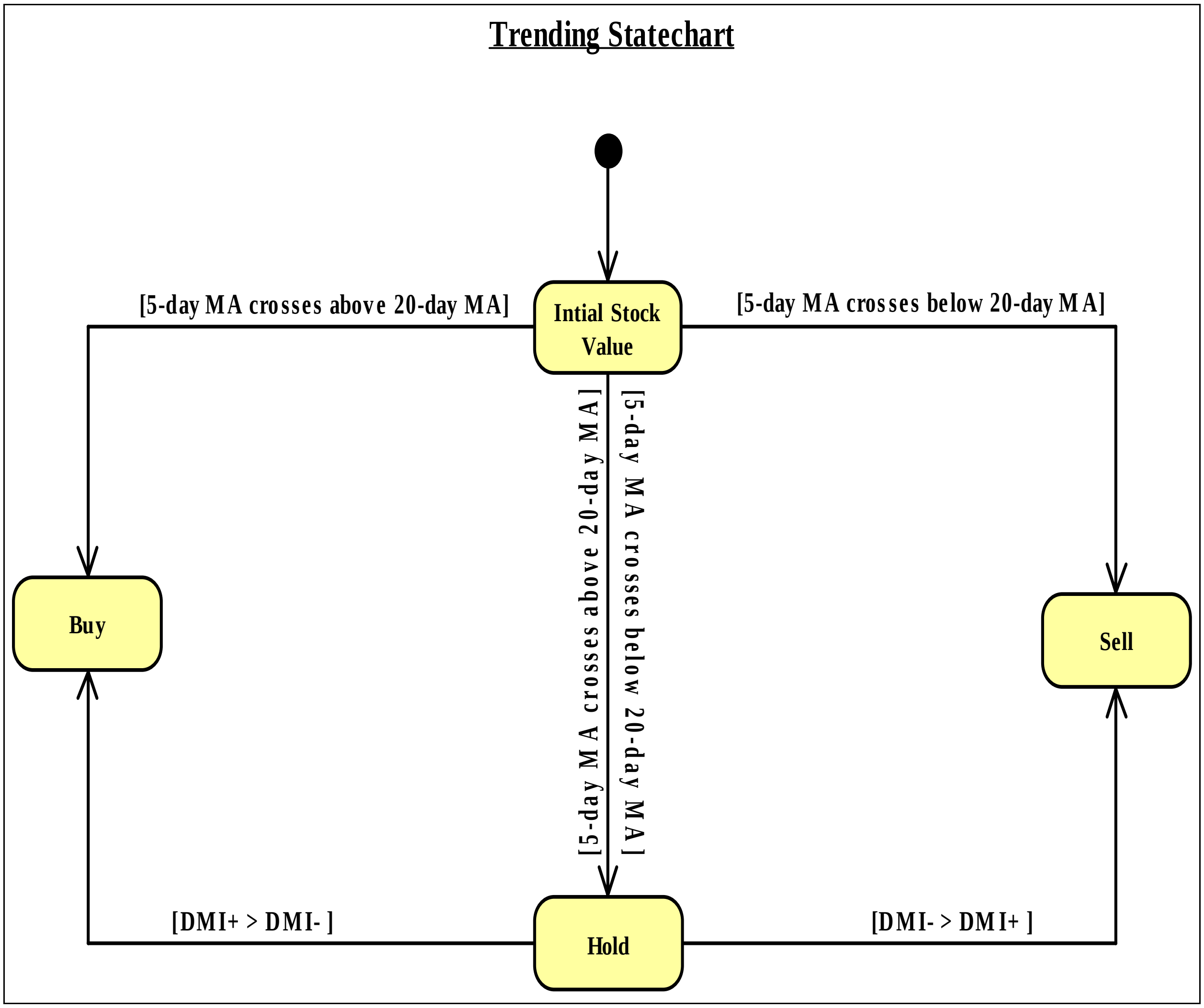

Trending Statechart
Intial Stock Value
[5-day MA crosses above 20-day MA]
[5-day MA crosses below 20-day MA]
Buy
Sell
[5-day MA crosses above 20-day MA]
[5-day MA crosses below 20-day MA]
Hold
[DMI+ > DMI-]
[DMI- > DMI+]

Figure17

5-SystemModel

In this section we bring the software model of our project.
As you remember, we know that we 3 different processes:
1- Market analysis
2- Optimization
3- Determining the best number of shares for each stock should be bought or sold.

Here we bring one model for the market analysis process and another model including the optimization and determination of the best number of shares.

5-1 Market Analysis Model:

The Market Analysis Model makes use of 5 basic indicators - the Average Directional Movement Index (ADX), moving average crossovers, stochastic oscillator, Bollinger bands, and Directional Movement Indicator (DMI) to determine and decide if a particular stock needs to be bought, sold, or simply on hold until the right time. Below is an illustration of each of these indicators showing how they operate and how they are used to make the correct decisions.Using this decision from the market the user can then make appropriate moves and make the most return on a given risk using the optimization model described in the next section.

5-1-1 ADX Indicator

ADX indicator helps determine whether a particular security is moving in a trending mode or an oscillating mode. This is a tremendously useful indicator, not because it tells you when to buy or sell (which it doesn't), but because it tells you which type of other indicator to use in a particular situation. For instance, in trending mode prices climb or drop in a pattern over time. The ADX responds with an increasing "trend" value.

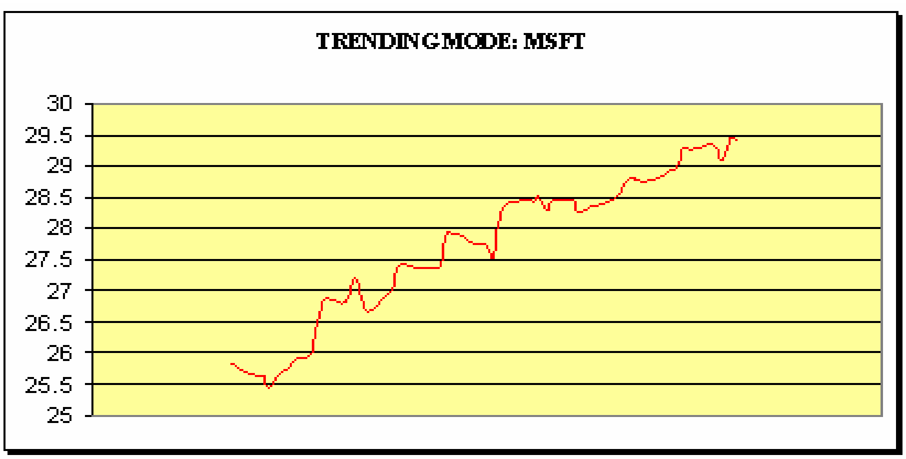


Figure18

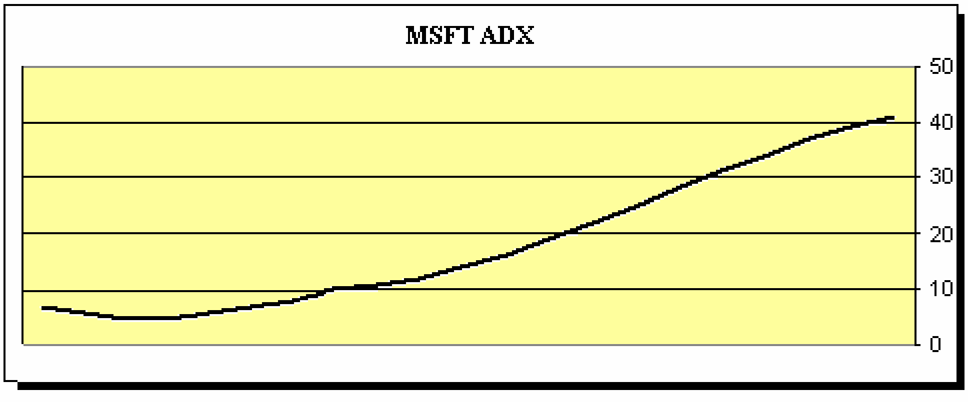


Figure19

As it can be seen from the graph of MSFT above, there is a clear upward trend in price movement from left to right. As the trend continues, the ADX value increases, showing the build-up in momentum until the trend itself begins to flatten.
When a security is trending, the ADX shows a high directional value, telling us to use a trend-following indicator such as moving average crossovers to make trading decisions.

In comparison, oscillating markets tend to go in one direction for a short time, then reverse direction, and reverse again in the original direction, swinging back and forth with high frequency in a somewhat-measurable range. During oscillating periods, the ADX shows a low value, telling us to use an oscillator such as the stochastic indicator to make trading decisions.

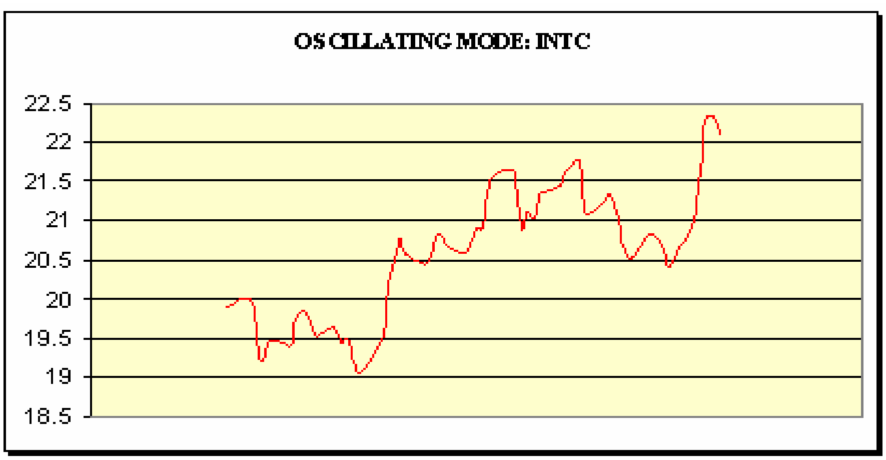


Figure20

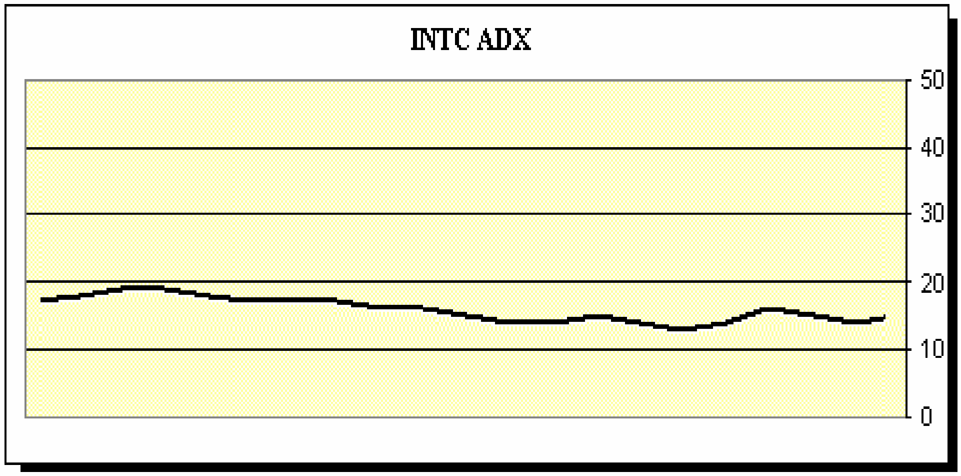


Figure21

By using the ADX to determine when a stock is "trending" versus "oscillating", we can then select either a trend indicator or oscillating indicator to use for our entry and exit signals. This rule is encompassed below:

RULE: If ADX exceeds 15, use moving average crossovers to generate buy/sell signals. If the ADX value is less than 15, use the stochastic indicator for buy/sell signals.

5-1-2 Moving Average – Trending

The most reliable indicator for profiting in trends is moving average crossovers. There are two ways to use moving average crossovers. First, you can use a single moving average and buy or sell when the price line crosses it. The second method is to use two moving averages, with the shorter moving average (such as 5 days) crossing over the longer moving average (such as 20 days) to give a signal. Below, we use a very short moving average as a proxy for price with a longer moving average in order to reduce lag and minimize false trades at the same time. For the moving average calculations we use the following rule:

RULE: When ADX is greater than 15, buy when 5 day moving average crosses above 20 day moving average, and sell when 5 day moving average crosses below 20 day moving average.

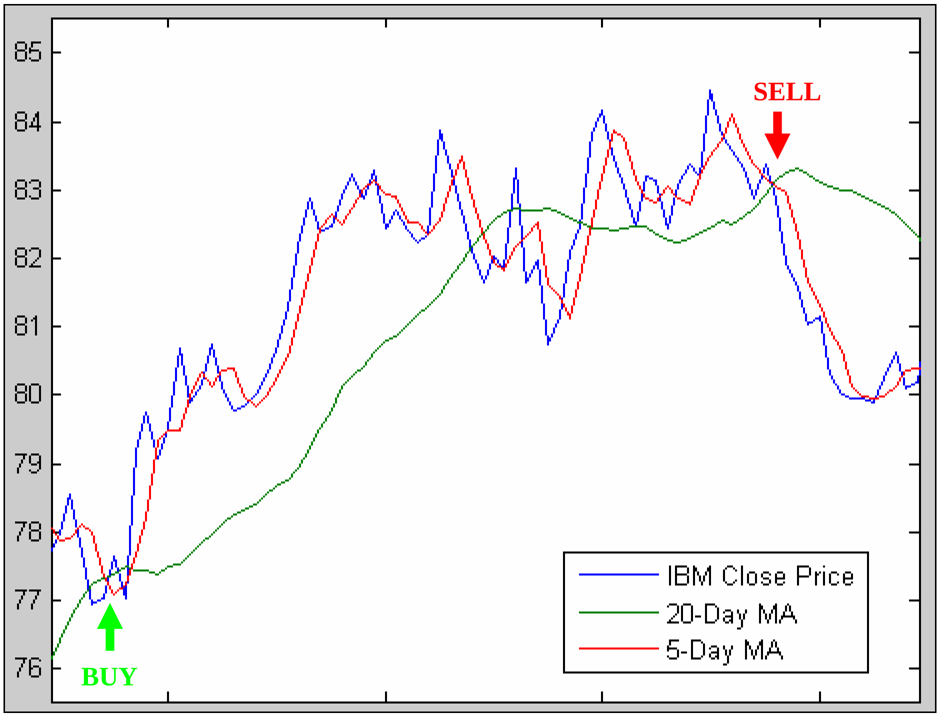


Figure22

5-1-3 Stochastic Oscillator

The stochastic oscillator is an indicator that does an excellent job of signaling overbought and oversold conditions and detecting changes in momentum at price tops and bottoms. The indicator can be applied to different time periods to increase sensitivity or decrease false trade signals.

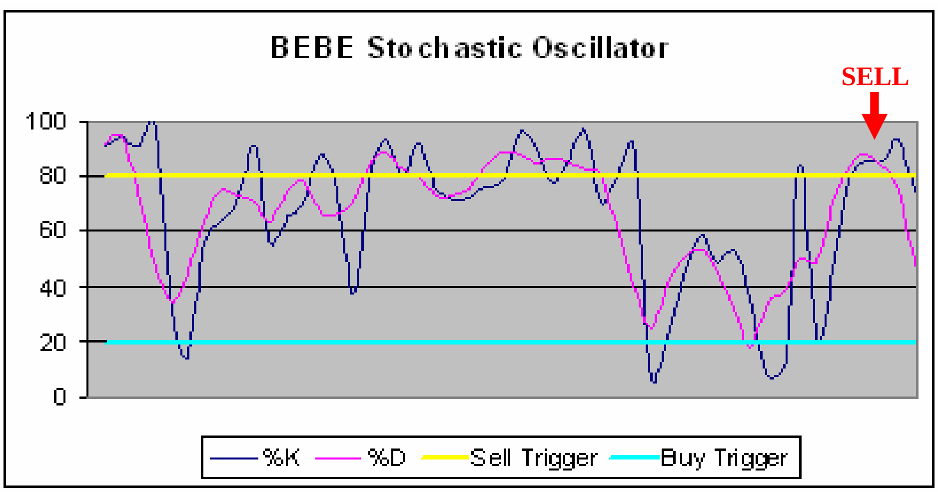


Figure23

There are two main components to the stochastic indicator which appear as oscillating lines that cross over one another. These are the %K line and the %D line. The %K line is the faster-moving, more sensitive version (think of a short-term moving average), while the %D line is just the moving average of the faster %K line. The %K line can vary from 5 to 10 periods, and the %D is generally a 3-period moving average of %K. When both lines are above 80, this indicates an overbought situation, and you should be looking to enter a short position or sell out a long position. When both lines are below 20, this indicates an oversold situation, and you should be looking to enter a long position or cover a short position. The actual signal comes when the fast %K line crosses the slower %D line, just like a moving crossover signal for trends. Therefore, the following rule is applied when the system performs the stochastic process:

RULE: When ADX is less than 15, then: Sell if both %K and %D are above 80 and %K crosses below %D; and Buy if both %K and %D are below 20 and %K crosses above %D.

5-1-4 Bollinger Bands

Bollinger bands create price envelopes that encapsulate most of the prices in a security's trading range at any given time. Bollinger bands do this by combining a self-adjusting volatility measure (standard deviation) with a moving average. The indicator uses a 10, 20, or 50-day moving average as the center line, and upper and lower bands calculated by adding to and subtracting standard deviations from the center line. Using two standard deviations, 95.5% of the prices over a given period should be contained within the upper and lower bands. Using one standard deviation, 66% of the prices are contained within the bands. Because only a small percentage of prices move outside the Bollinger bands when there is a price extreme, this indicator is very useful in timing trades to capture tops and bottoms during both trending and oscillating periods. The use of Bollinger bands can increase profits over time and reduce risk and trading costs by eliminating large number false trades.

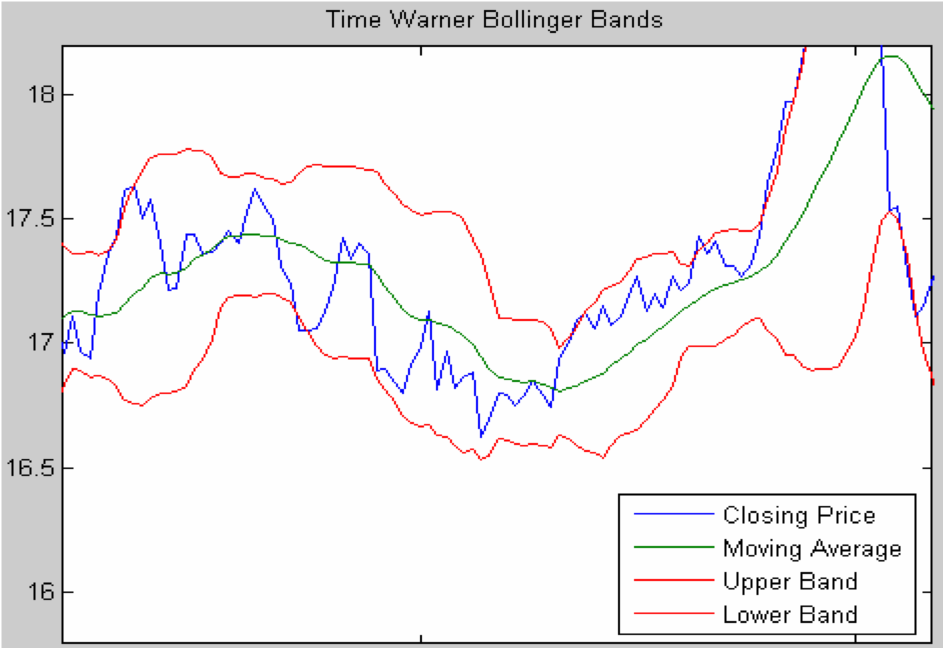


Figure24

Like the ADX, Bollinger bands do not actually give trading signals. Rather, they enhance and confirm the moving average crossovers and stochastic oscillator. When prices touch or penetrate the upper band, this signals an overbought condition and we should be looking to sell. Likewise, when prices touch or penetrate the lower band, this signals an oversold condition and we should be looking to buy.

Now that we understand how different types of indicators operate and what the system should utilize to make the correct buy/sell/hold decisions for the user, we can now build a model using the methods and rules mentioned above.

The system is extensively built using Microsoft Excel and Visual Basic. The Visual Basic code can be seen in the appendix below. Basically this is how the system is modeled:

When the user initiates the application this is what he/she will see.

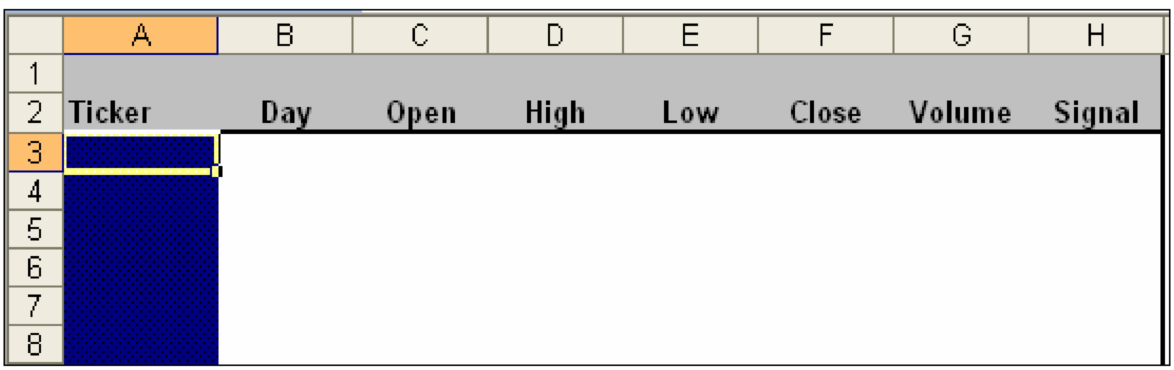


Figure25

The system at this point is waiting for the user's input. The user enters the tickers or stocks he/she would like to purchase and invest in just as shown below:

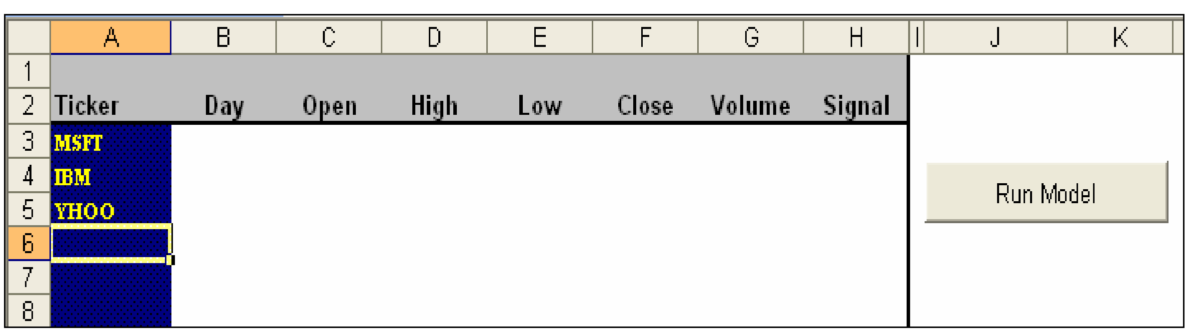


Figure26

At this point all the calculations are null or no calculations have been carried out:

	A	B	C	D	E	F	L	M	S	U	V	Z	AA
1													
2	Label	Open	High	Low	Close	Volume	DMI14+	DMI14-	ADX	5 Day	20 Day	%K	%D
3							0	0	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!	#DIV/0!
4							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
5							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
6							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
7							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
8							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
9							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
10							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
11							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
12							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
13							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
14							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
15							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
16							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
17							0	0	#DIV/0!		#DIV/0!	#DIV/0!	#DIV/0!
18											#DIV/0!	#DIV/0!	#DIV/0!
19											#DIV/0!	#DIV/0!	#DIV/0!

Fiure27

When the user clicks "Run Model" the following actions and calculations are carried out immediately:

- The Data from the Database is pulled for each stock that the user wants to invest in.

Tickers

GOOG						GTW						IBM	
17-Nov-06	493.25	499.66	493	498.79	5511000	17-Nov-06	1.84	1.92	1.82	1.91	3424800	17-Nov-06	93.42
16-Nov-06	495	497.68	492.56	495.9	5092600	16-Nov-06	1.89	1.9	1.82	1.84	2560300	16-Nov-06	93.07
15-Nov-06	493.43	499.85	491.93	491.93	8370700	15-Nov-06	1.91	1.93	1.88	1.88	4366200	15-Nov-06	93.08
14-Nov-06	480.7	489.95	480.5	489.3	7223400	14-Nov-06	1.91	1.93	1.85	1.91	5725500	14-Nov-06	92
13-Nov-06	474.9	481.17	474.14	481.03	4341900	13-Nov-06	1.73	1.86	1.73	1.85	6913900	13-Nov-06	91.55
10-Nov-06	473.78	474.72	470.29	473.55	2796700	10-Nov-06	1.7	1.78	1.7	1.73	4019700	10-Nov-06	92.3
9-Nov-06	476.5	479.49	471.86	472.63	4879200	9-Nov-06	1.65	1.71	1.65	1.7	4780300	9-Nov-06	92.59
8-Nov-06	470.35	481.74	468.6	475	7965000	8-Nov-06	1.6	1.65	1.6	1.63	3072800	8-Nov-06	92.25
7-Nov-06	476.95	479.02	471.77	472.57	4897100	7-Nov-06	1.63	1.64	1.6	1.6	5060600	7-Nov-06	92.42
6-Nov-06	473.77	479.66	472.33	476.95	4991500	6-Nov-06	1.62	1.68	1.59	1.6	5816500	6-Nov-06	91.5
3-Nov-06	472.23	473.75	465.06	471.8	4907700	3-Nov-06	1.73	1.74	1.6	1.64	6879200	3-Nov-06	91.76
2-Nov-06	467.5	473.73	466.38	469.91	5236700	2-Nov-06	1.65	1.66	1.62	1.64	2504500	2-Nov-06	91.8
1-Nov-06	478.76	479.13	465.26	467.5	5426300	1-Nov-06	1.68	1.7	1.65	1.65	1709400	1-Nov-06	92.5
31-Oct-06	478.06	482.16	473.84	476.39	6285400	31-Oct-06	1.67	1.7	1.65	1.68	3129600	31-Oct-06	91.5
30-Oct-06	474.82	480.46	470.01	476.57	6563100	30-Oct-06	1.67	1.7	1.65	1.66	2883000	30-Oct-06	89.8

AC DF GI JL MO PR SU VX YZ

Figure28

- ADX calculations to check to see if the movement of the stock is "Trending" or "Oscillating".
- Based on the ADX, the system either performs
  - Moving Average calculations – DMI and MA
  - Stochastic Calculations – Bollinger Band, %K and %D

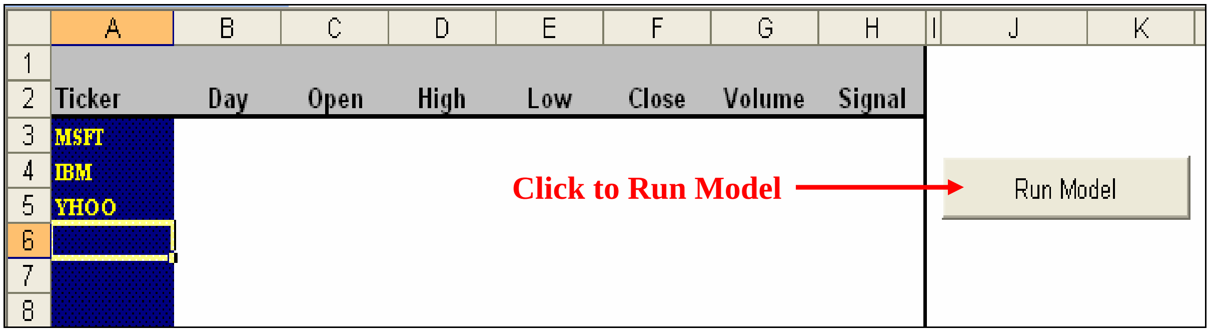


Figure29

The outputs to these calculations are – Buy/Sell/Hold – just as seen below:

Ticker	Day	Open	High	Low	Close	Volume	Signal
MSFT	39038	29.31	29.54	29.28	29.4	49356700	Buy
IBM	39038	93.42	94.05	93.31	93.81	5251000	Buy
YHOO	39038	26.68	27.05	26.63	26.91	17955200	Hold
BEBE	39038	23.88	23.91	23.12	23.5	820500	Sell

Run Model

Figure30

The associated calculations are:

Label	Open	High	Low	Close	Volume	DMI14+	DMI14-	ADX	5 Day	20 Day	%K	%D
39038	23.9	23.9	23	23.5	820500	1.812	3.77	27	23.59	24.2505	59.167	77.419
39037	23.9	24	24	23.88	569500	1.952	3.361	27	23.592	24.3215	90.833	86.472
39036	23.8	24	24	23.77	461800	2.103	3.62	27.154	23.408	24.3685	82.258	61.177
39035	23.2	23.9	23	23.76	1E+06	2.114	3.899	27.25	23.332	24.4575	86.325	46.816
39034	23.6	23.8	23	23.04	700900	2.169	4.199	26.909	23.442	24.542	14.948	21.649
39031	23	23.6	23	23.51	604200	2.121	4.522	26.5	23.69	24.6955	39.175	27.072
39030	23.5	23.6	23	22.96	869700	2.285	4.87	25.333	23.694	24.8045	10.825	35.524
39029	23.5	23.9	23	23.39	2E+06	2.461	5.148	23.875	23.956	24.933	31.217	52.998
39028	24.3	24.7	24	24.31	1E+06	2.651	4.446	22.143	24.12	24.9995	64.53	52.991
39027	23.7	24.5	24	24.28	591100	2.619	4.789	21.667	24.224	25.0325	63.248	52.422
39024	24.4	24.5	23	23.53	632400	2.821	5.158	20	24.304	25.067	31.197	52.93
39023	23.1	24.5	23	24.27	933000	3.017	5.555	17.5	24.454	25.1405	62.821	60.345
39022	25.1	25.1	24	24.21	814200	3.25	4.583	13.667	24.438	25.1705	64.773	56.071
39021	24.9	25	24	24.83	1E+06	3.501	4.559	12.5	24.944	25.238	53.44	48.089

Figure31

As mentioned above, using this output the user can make a sound decision on whether to buy, sell, or hold a particular share. The next thing to do is determine the amount of shares to buy or sell and then based on that make the best optimal portfolio. This is seen below in the next section.

5-2 Optimization model

As we have already discussed one of the important processes in our system in optimization. We assume that this system do the optimization just once by beginning of the investment period to calculate the optimum weight of these stocks in the portfolio. By passing every day we do market analysis independently for each stock to see whether we should buy, sell or hold stocks, then by means of new closed price and obtained optimum stock's weights from optimization, we can say how many share of each stock we should buy or sell

5-2-1 optimization mathematical model

Notation:

R_i : The annual returns of stock i from historical data (i=1…n)

$\overline{R}_i$: The *Average* annual returns of stock i from historical data (i=1…n)

$P_i(t)$: The closing price of stock i at time t (at the end of day t) (i=1…n)

δ_i : The average standard deviation of return for stock i form historical data (i=1…n)

W_i : The weight of value of stock i in our portfolio (i=1…n)

B : The initial amount of investment we assume it's fixes throughout the investment period.

M : The Minimum expected of return for investor

$N_i(t)$: The number of shares of stock i at time t (i=1…n).

Z: The volatility of risk of the portfolio

Upperbound: is the maximum of weight you want for that stock
Lowerbound: Is the minimum of weight you want for that stock

Optimization model:

The objective function:

Min $Z=\sqrt{\sum_{i=1}^{n} W_i^2\delta_i^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} W_i W_j Cov(R_i, R_j)}$

S.T.

$Lowerbound_i \le W_i \le Upperbond_i$ For (i=1…n)

$$\sum_{i=1}^{n} W_i \overline{R_i} >= M$$

$$\sum_{i=1}^{n} W_i = 1$$

After we obtained the minimum of Z and also the best weights, we can obtain the $N_i(t)$ by having the $P_i(t)$ by the end of the each day.

Formula for determining the best number of shares by knowing sell or buy decision from market analysis process is:

$$N_i(t) = \frac{W_i B}{P_i(t)}$$

5-2-2 Optimization and determining the amount of shares to buy or sell (excel model)

To explain the model we bring an example we are going to make use of that in the verification section.

As it is understandable from the mathematical model each investor should provide the system with the some initial information such as the investment amount , the lower and upper bound of the preferred weights for each stock and the minimum expected return.
The other inputs such as historical annual return needed for the optimization model obtain from the stock's data base.

The optimization's objective function is to minimize the risk and standard deviation of the portfolio for the given minimum expected return.

The excel model consists of the 2 sheets.
The first one is the input sheet which the investor should give the system asked information.

Assume that we want to have 3 the shares of the 3 stocks such as Yahoo, IBM and Microsoft. We know that we desire to invest 1000$ in a portfolio consisting of these 3 with different allocations.
Additionally we want to have at least 30 % of return by the end of the investment period and we are intended to put some restrictions on the weight of each stock as shown in figure32.

In the figure 32 we have historical information just for 10 periods in regard to their average of the return.
The figure 32 shows the input sheet for the afore-mentioned example.

Budget:	$1,000		Exp. Return:	30.00%
Min Constraint:	10.00%	10.00%	10.00%	
Max Constraint:	100.00%	70.00%	100.00%	
Ticker:	MSFT	IBM	YHOO	
Period 1	47%	78%	16%	
Period 2	24%	78%	17%	
Period 3	24%	78%	16%	
Period 4	26%	85%	18%	
Period 5	25%	88%	21%	
Period 6	26%	83%	21%	
Period 7	26%	81%	25%	
Period 8	27%	82%	29%	
Period 9	28%	88%	28%	
Period 10	26%	89%	33%	

Figure32 (Input sheet)

After we fill the input sheet completely we will go to the second sheet in excel model called result sheet.
In this sheet we have optimization command which can solve the optimization problem and calculate the optimum weights of each stock to have the minimum of the portfolio risk for the given expected return.
Some times, we can have more than expected return for the minimized calculated portfolio risk.
If we solve the optimization problem, for our example, we have the results in the result sheet shown below.
Figure 33 demonstrates the result sheet for this example.

Exp Return:	30.00%	Prod:	3
Max Return:	57.36%	StdDev:	3.146%

Optimize

Stock	Curr. Close Price	Weights	Min	Max	Returns	Variance	No. of Shares	MSFT	IBM	YHOO
MSFT	$26.91	32.1%	10.0%	100.0%	27.81%	0.005	11.94	0.44%	-0.08%	-0.10%
IBM	$73.08	54.7%	10.0%	70.0%	83.11%	0.002	7.49	-0.08%	0.17%	0.16%
YHOO	$20.43	13.1%	10.0%	100.0%	22.32%	0.003	6.42	-0.10%	0.16%	0.31%

Figure33 (result sheet)

As you can see here by this optimization we obtain the optimum weights for each stock resulting in having best portfolio with minimum risk while satisfying the expected return.

In this example we can have the return of the 57.36% instead of expected 30 % with the same risk.
For this example the optimum weights are shown in the figure 34.

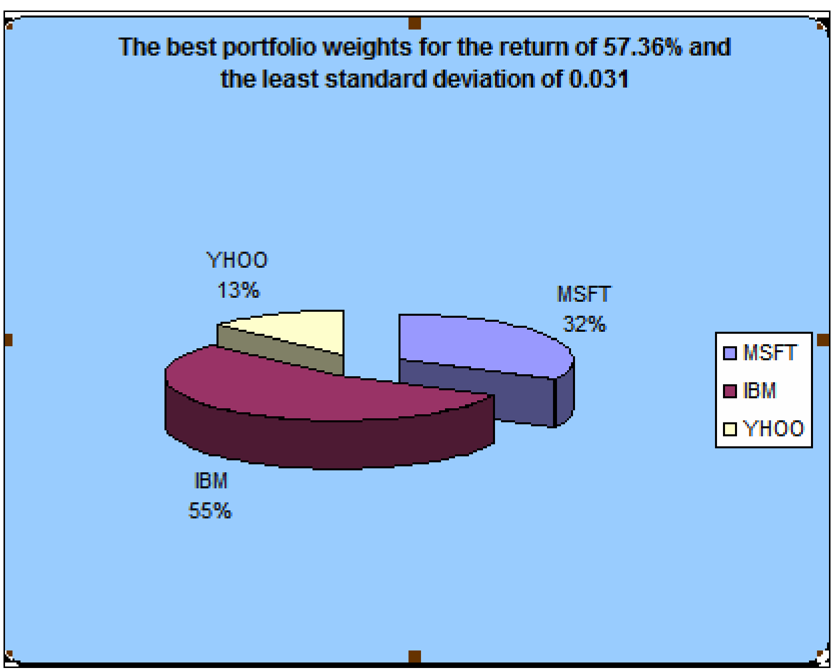


Figure34

As soon as you obtain the optimum weights for each stock, by having the closing price of that day for stocks and also the buy , hold or sell decision from the market analysis , we have obtain the best number of shares for each stock in the next day.

By calculating the best number of shares by $N_i(t) = \dfrac{W_i B}{P_i(t)}$, we can say by obtaining the $N_i(t) - N_i(t-1)$ how many shares should buy or sell. And these are the best number of but and sell since can keep the optimum expected return and risk at the same optimum level obtained initially by the optimization process.

For our example, if we determine the best number of shares for the next day, based on the optimum weights and also the latest closing price, the results will be shown in the figure 35.

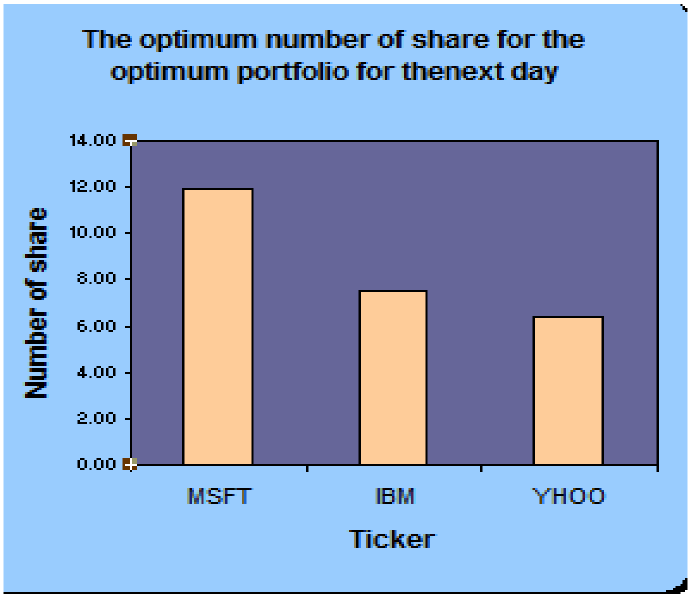


Figure 35

5-2-3 Optimum line

The optimum line is a line manifesting the best possible points possessing the potential maximum portfolio return versus the minimum possible portfolio risk (standard deviation).
In other words, this line can show that for any amount of portfolio risk existing what is the best possible maximum return.

If we solve the optimization problem for our example for all of the minimum expected return larger than zero, we can generate this line.

Figure 36 demonstrates this line for our example.

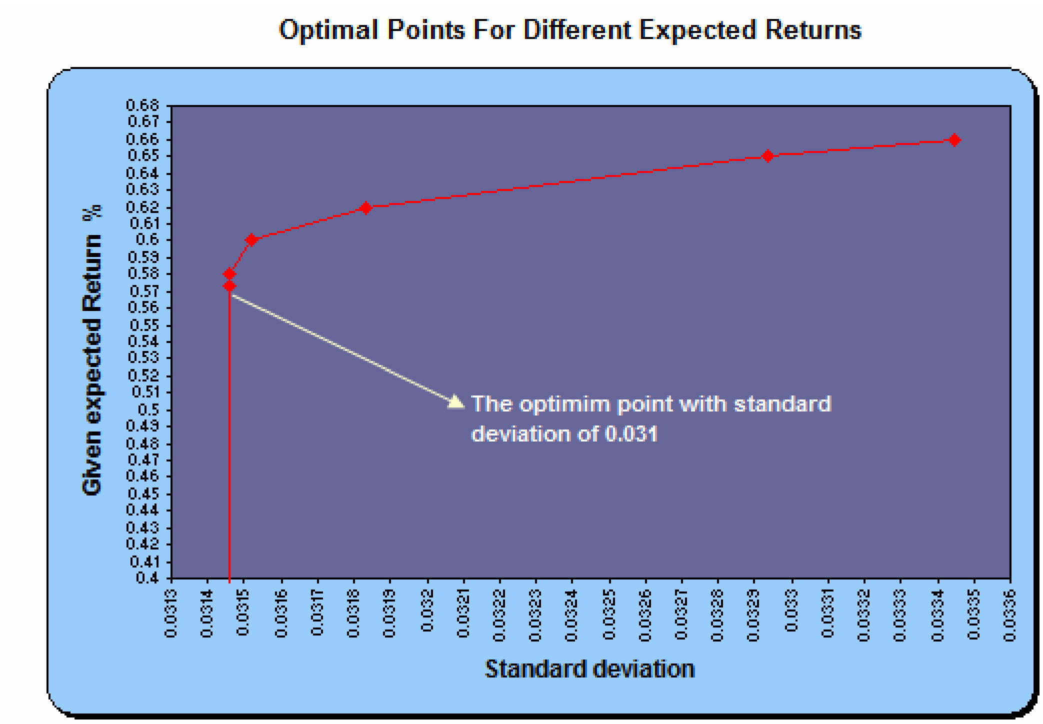


Figure37

You can see the optimum solution of our example when you desire to have at least 30 % returns. You can see for the same risk there is a point that can have the return of 57%.
So this point is our optimum solution when you expected return as an initial input is between 0 and 57%.

We will also make use of the results of the optimum line for this example in the verification section.

6-Validation and Verification

Once the system software has been developed, it is time to validate and verify, in order to ensure that it performs as required. Moreover, Validation means that we verify the .as described. Behavior corresponds to what we want to specify. Verification through simulation can be carried out by using so-called test patterns. as inputs to the system, and verifying that the outputs are as required.
Since our 2 most important processes are market analysis and optimization, we have the verification for theses 2.

6-1 Validation

Based on the initial set of requirements above and based on the system model, we can conclude that the system is validated properly. Here are some of the requirements and associated validations:

Requirements	Validation
The system should be designed and manipulated in a user-friendly format which most people knows a little a bit about it.	The entire model is based on Microsoft Excel, which means it is easy to use and readily available for anybody that has a computer. Furthermore, since it uses VBA to manipulate data, any user can further help develop the model and take it to the next level.
The system should make use of the decision results from market analysis in order to calculate the best number of the buy, sell of each stock.	The system automatically gathers data from the "data" sheet without any need for manual input from the user. The user only has to provide the stocks or securities he/she would like to invest in. With just one push of a button, the system is capable of giving the user the best buy/sell/hold analysis from which the user can make key decision on his/her investment. The user does not have to go through the day monitoring what the market is doing or have the headache of making a critical decision.
The system should construct the minimum optimization model to minimize the risk or overall standard deviation of the portfolio for a given expected return, upper and lower bound of each stock's weights.	When the optimization model is run the system not only calculates the number of shares a user must invest in, the system also, based on the risk the user wishes to allocate, performs weight calculations, covariance calculations, and optimizes the entire portfolio such that the user always gets the maximum return. It also in his/her next investment levels the portfolio in such a way

	that the investor never loses money. He/she either has a profitable portfolio or has the ability to maintain the set of portfolio.
The system should calculate the best the number of buy or sell shares for each stock after obtain the latest closing price of each stock and the results from market analysis.	The system, based on the calculated weights, the closing price, and the budget, gives the user the exact number of shares to buy or sell. This is a unique attribute the system adds to this model. Now the user can peacefully buy or sell shares from the market analysis and also buy or sell exact number of shares to make the most profit.
The investor should be informed about content of the report in order to understand the results perfectly for implementing the daily decision.	The system not only displays the exact amount of profit the user can make it also displays a series of charts which the user can easily visualize and interpret. This again adds to the usability and durability of the system.

Table6

To verify that the product was built right we have used Matlab extensively. Matlab is the best possible tool to perform verification since it can not only perform complex data analysis, it is also a great supplement to excel.

6-2 Verification

6-2-1 Market Analysis Verification

The verification of buy and sell can very easily be done in Matlab. Matlab offers several different functions that can be utilized directly to perform financial analysis. Here we use matlab to draw a moving average model and Bollinger model and also determine whether we buy or sell a particular stock. Since this is an example set, we will use IBM as the main stock to verify our model.

The first thing we need to do is check the stock volatility. We will use Relative Strength Index and ADX to see which type of indicator to use. RSI code is in the appendix. Here we will discuss the output.
When RSI is run, we see:

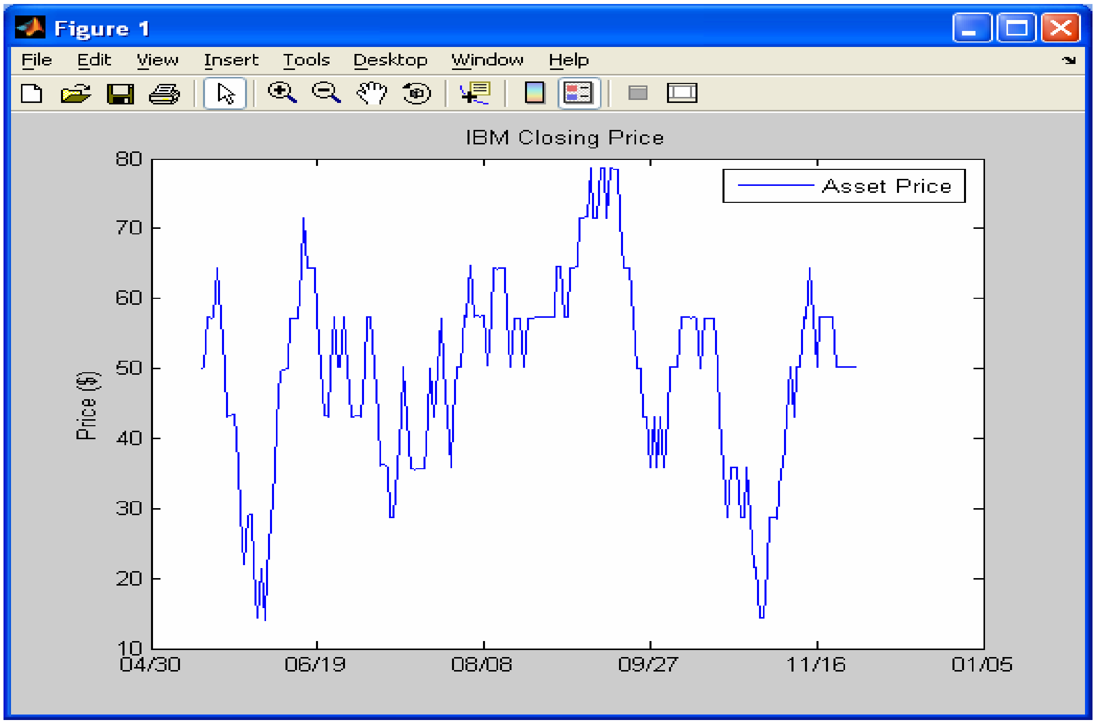


Figure38

Now, looking at the ADX we see:

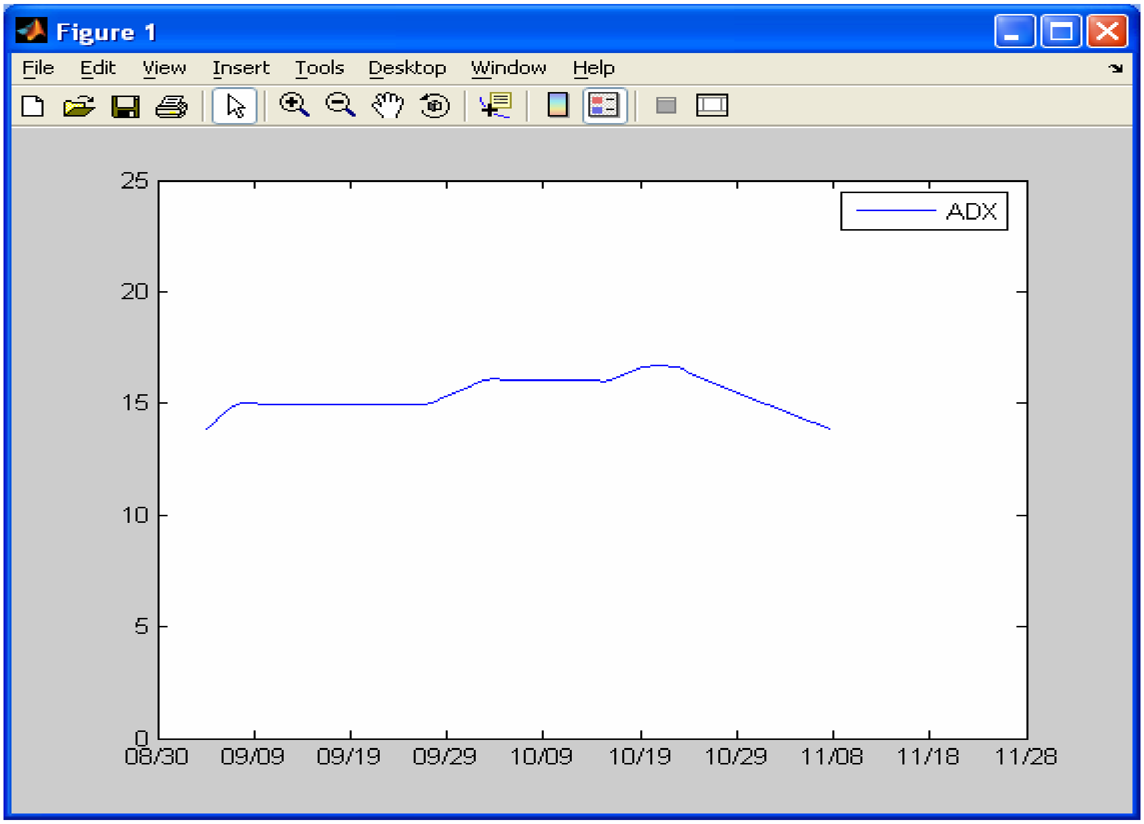


Figure39

This tells us that the Stock is volatile and confirms with our results from above calculations. We know now that we need to use the stochastic process with Bollinger Band. We will use the "bolling" function given by the Matlab program to figure this. The code is again in the appendix.

Using bolling function we get:

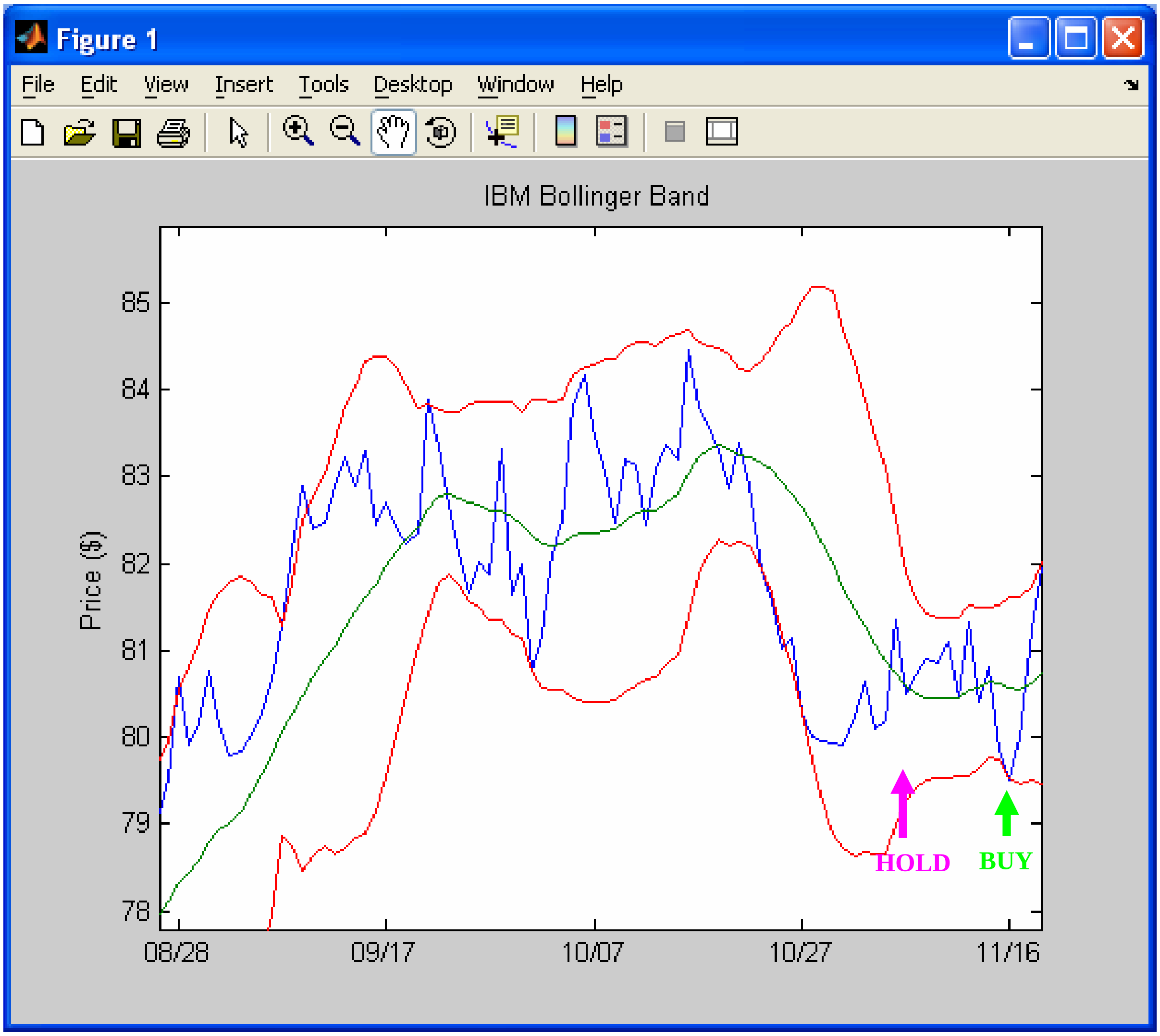


Figure40

This again confirms the decision we saw earlier for IBM stock – BUY.

6-2-2Verification of Optimal Portfolio (Optimization)

In the optimization modeling above we used a set of rules, mathematical calculations, and Excel and VBA to come up with an optimal portfolio for the user. To verify that the system indeed made the right choice of providing an optimal portfolio, we make use of the concept called Efficient Frontier.

The Efficient Frontier in financial models solves the problem of finding portfolio with minimal variance or risk for a given expected return, or, (nearly) equivalently, with maximal expected return for a given risk. In our model, we solve the Efficient Frontier problem using Matlab so we can verify that the system that has been built has made the correct choice and given the user the most optimal portfolio. We make use of built-in functions in Matlab to build an Efficient Frontier. If the results are correct then we should see that the portfolio for the given risk and return lies at the Efficient Frontier Curve. Therefore, solving for it, we get:

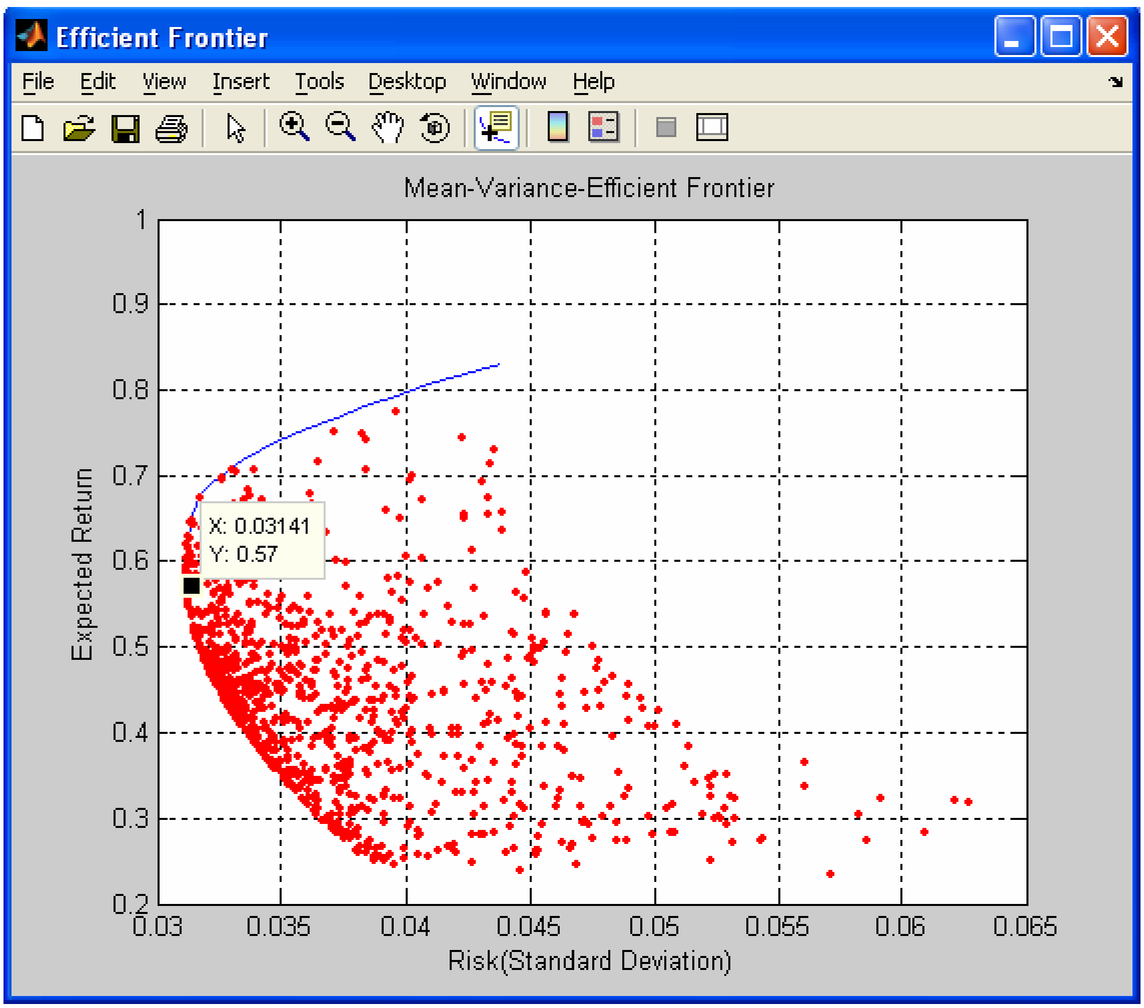
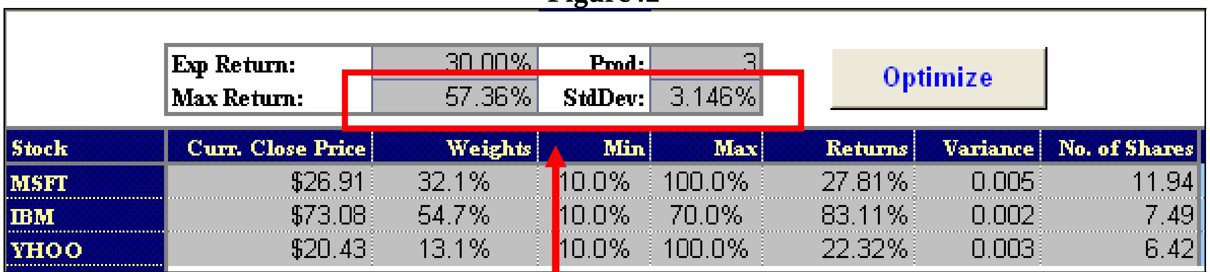


Figure41

From the results above, for the given risk of 3.1% we see a return of 57%, which corresponds exactly to the results we obtained when we ran the optimization program in excel, seen below.

Figure42



Exp Return:	30.00%	Prod:	3
Max Return:	57.36%	StdDev:	3.146%

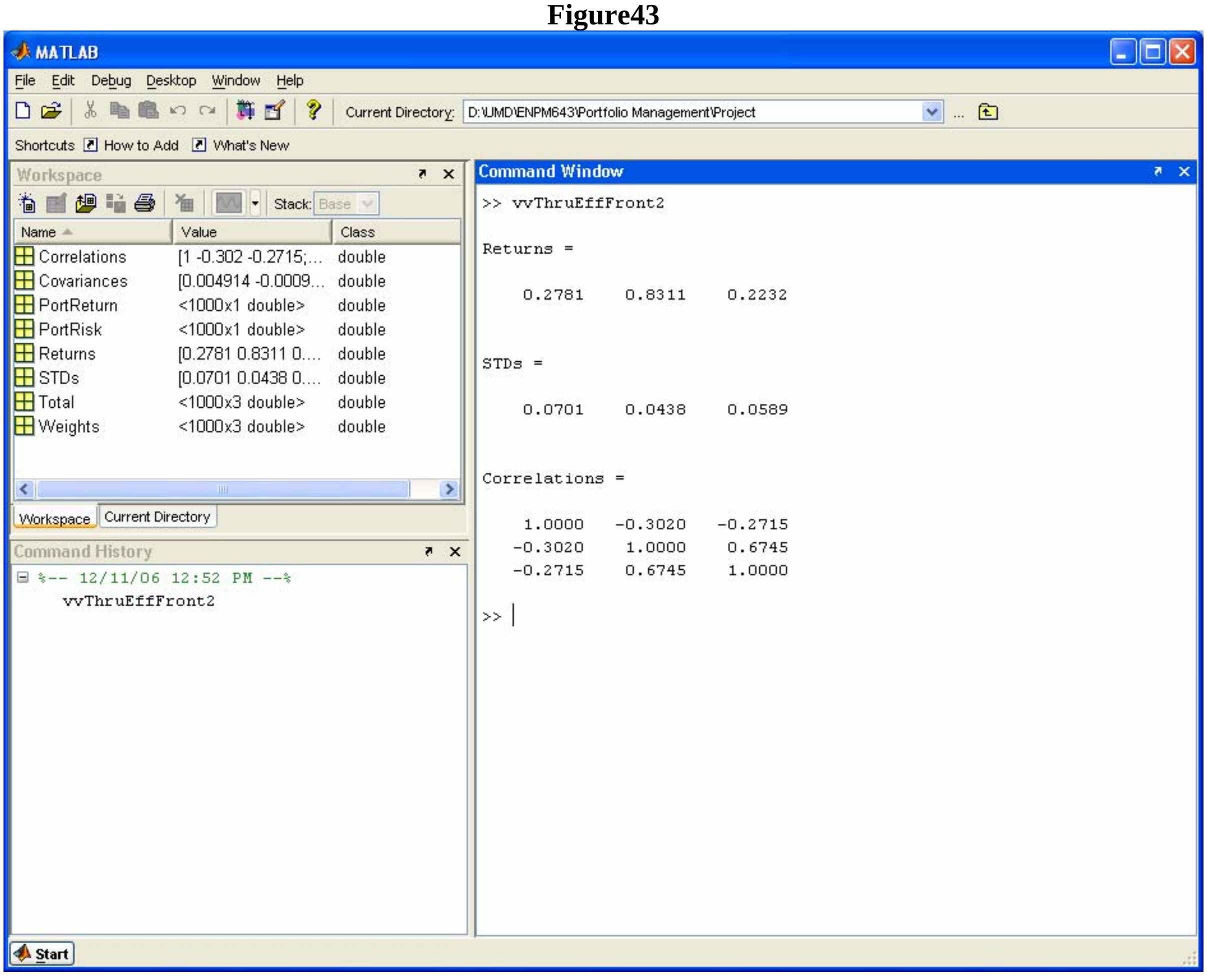
Optimize

Stock	Curr. Close Price	Weights	Min	Max	Returns	Variance	No. of Shares
MSFT	$26.91	32.1%	10.0%	100.0%	27.81%	0.005	11.94
IBM	$73.08	54.7%	10.0%	70.0%	83.11%	0.002	7.49
YHOO	$20.43	13.1%	10.0%	100.0%	22.32%	0.003	6.42

Maximum Return for a given Risk

This verifies and confirms that the user will have the most optimal portfolio if he/she follows what the system is given them. The corresponding calculations are seen below:

Figure43



7-Summary

In this project we can construct the best portfolio with the minimum risk and maximum return among bunch of the stocks. This system can guarantee that if you conform to having the optimum obtained stock's weights at all time, you can achieve your goal which is having an optimum portfolio in the end of the investment period.

To put it bluntly, after the optimization is run for the first time, according to the initial investor's preference, we obtain the best weights for each stock. Then, having the new closing price and knowing the results of daily buy, sell or hold decision from market analysis for each stock, we can calculate how many shares of each stock we should trade (buy or sell) to be sure that we are still on the right track .

To understand the sequence and the relationship between optimization and market analysis process and how they simply work we show how behavioral modeling can be transformed into a functional software design in VB and Excel in the modeling section.

Simulation is the validation tool used in this project to verify that the behavior expected from optimization is actually correspondent to our objective. By running the MATLAB-generated simulation we can verify that results of our random-generated simulated system's behaviors in terms of the efficient frontier are so close to the outcomes of the optimization. We did the similar approach to verify the behavior of the market analysis to make sure buy, sell and hold decision are nearly achieved according to the real financial market.

The work in this report was initially so laborious, since both authors knew a little about the financial market and had a preliminary prospective because of the new topic has been for this project. Many of the requirements and lessons learned from studied many papers and online recourses have been used to firstly design this system institutively and secondly analyze as well as verify this system.

Both authors of the report have learned many fundamental systems engineering techniques and existing concepts in the real financial world, which will be valuable for use elsewhere in real life systems.

Appendix

1-Appendix for Figure 6:

Before any further explanation in regard to stocks and the decision making policies for how allocate your capital among the list of some chosen stocks you should know how to read a stock table.

Here we have explanations just for columns that we have used in your system modeling both in the market analysis and optimization processes.

Columns 1 & 2: 52-Week High and Low - These are the highest and lowest prices at which a stock has traded over the previous 52 weeks (one year). This typically does not include the previous day's trading.

Column 3: Company Name & Type of Stock - This column lists the name of the company. If there are no special symbols or letters following the name, it is common stock.

Column 4: Ticker Symbol - This is the unique alphabetic name which identifies the stock.
Column 8: Trading Volume - This figure shows the total number of shares traded for the day, listed in hundreds. To get the actual number traded, add "00" to the end of the number listed.

Column 9 & 10: Day High and Low - This indicates the price range at which the stock has traded at throughout the day. In other words, these are the maximum and the minimum prices that people have paid for the stock.

Column 11: Close - The close is the last trading price recorded when the market closed on the day. If the closing price is up or down more than 5% than the previous day's close, the entire listing for that stock is bold-faced. Keep in mind, you are not guaranteed to get this price if you buy the stock the next day because the price is constantly changing (even after the exchange is closed for the day).

2-Appendix for the market analysis modeling codes

2-1 VBA Code for Market Analysis:

Main Function – Running the Model

```
Sub Run_Model()
   Call Signals
   Sheets("Signals").Select
End Sub
```

Function – Signals

```
Sub Signals()
Dim rowcount, colcount, datacol, datarow As Integer
Dim ticker, ADX, MA5, MA20, StochK, StochD, BollUpper, BollLower As Variant
Dim DMIup, DMIdn, price, signal As Variant

-- Activate Workbook - -
   Workbooks("Model.xls").Activate
      Sheets("Signals").Select

-- Select the Range of Data for each stock --
      For rowcount = 1 To Range("Data").Rows.Count
            If Range("Data").Cells(rowcount, 1).Value = "" Then
               Exit For
            ElseIf Range("Data").Cells(rowcount, 1).Value <> "" Then

-- Assign ticker variable and select the data from DATA.xls --
               ticker = Range("Data").Cells(rowcount, 1).Value

-- Get stock data from DATA.xls workbook --
               If ticker = "" Then
                  Exit For
               Else:
                  Workbooks("DATA.xls").Activate

-- Select correct sheet to get price data --
                  If Left(ticker, 1) = "A" Or Left(ticker, 1) = "B" Or _
            Left(ticker, 1) = "C" Then
            Sheets("AC").Select
                  ElseIf Left(ticker, 1) = "D" Or Left(ticker, 1) = "E" Or _
            Left(ticker, 1) = "F" Then
            Sheets("DF").Select
                  ElseIf Left(ticker, 1) = "G" Or Left(ticker, 1) = "H" Or _
```

```
            Left(ticker, 1) = "I" Then
            Sheets("GI").Select
                ElseIf Left(ticker, 1) = "J" Or Left(ticker, 1) = "K" Or _
            Left(ticker, 1) = "L" Then
            Sheets("JL").Select
                ElseIf Left(ticker, 1) = "M" Or Left(ticker, 1) = "N" Or _
            Left(ticker, 1) = "O" Then
            Sheets("MO").Select
                ElseIf Left(ticker, 1) = "P" Or Left(ticker, 1) = "Q" Or _
            Left(ticker, 1) = "R" Then
            Sheets("PR").Select
                ElseIf Left(ticker, 1) = "S" Or Left(ticker, 1) = "T" Or _
            Left(ticker, 1) = "U" Then
            Sheets("SU").Select
                ElseIf Left(ticker, 1) = "V" Or Left(ticker, 1) = "W" Or _
            Left(ticker, 1) = "X" Then
            Sheets("VX").Select
                ElseIf Left(ticker, 1) = "Y" Or Left(ticker, 1) = "Z" Then
                    Sheets("YZ").Select
                End If
```

-- Go across top of sheet looking for ticker --

```
                For datacol = 1 To Range("a1:iv1").Columns.Count Step 6
                  If Range("a1:ff1").Cells(1, datacol).Value = "" Then
                    Exit For
                  ElseIf Range("a1:iv1").Cells(1, datacol).Value = ticker Then
```

-- Copy data --

```
                    Range(Cells(2, datacol), Cells(101, datacol + 5)).Copy
                    Exit For
                  End If
                Next datacol
```

-- Return to Model.xls --

```
              Workbooks("Model.xls").Activate
              Sheets("Calculations").Select
```

-- Copy to reversal area and sort descending by date --

```
              Range("am3").Select
                Selection.PasteSpecial Paste:=xlValues
                Selection.Sort Key1:=Range("AM3"), Order1:=xlDescending, _
          Header:=xlGuess, OrderCustom:=1, MatchCase:=False, _
          Orientation:=xlTopToBottom
                Selection.Copy
```

-- Copy reversed data to calculation area --

```
                 Range("a3").Select
                  Selection.PasteSpecial Paste:=xlValues
              End If
```

-- Assign indicator variables --

```
                price = Range("E3").Value
                ADX = Range("S3").Value
                MA5 = Range("U3").Value
                MA20 = Range("V3").Value
                StochK = Range("Z3").Value
                StochD = Range("AA3").Value
                BollUpper = Range("AE3").Value
                BollLower = Range("AF3").Value
                DMIup = Range("AH3").Value
                DMIdn = Range("AI3").Value
```

-- Calculate signals --

```
                signal = ""
```

-- Trend signals with ADX, moving averages, and DMI --

```
                Sheets("Calculations").Select
                If ADX > 15 And MA5 > MA20 And DMIup > DMIdn Then
                  signal = "Buy"
                ElseIf ADX > 15 And MA5 < MA20 And DMIup < DMIdn Then
                  signal = "Sell"
```

-- Oscillating signals with ADX, Bollinger bands, and stochastics --

```
                ElseIf Range("Oscillating_Zone").Value <> "" And ADX < 15 And _
          price < BollLower And StochK < 20 And StochD < 20 And StochK > StochD Then
                  signal = "Buy"
                ElseIf Range("Oscillating_Zone").Value <> "" And ADX < 15 And _
          price > BollUpper And StochK > 80 And StochD > 80 And StochK < StochD Then
                  signal = "Sell"
                Else:  signal = "Hold"
                End If
```

-- Insert signal into \Signals/ sheet col H --

```
              Sheets("Signals").Select

                Range("Signal").Cells(rowcount, 1).Value = signal
```

-- Clear signal variable --

```
              signal = ""
```

-- Inserts latest data in \Signals/ sheet cols B:G --

```
              Sheets("Calculations").Select
                Range("A3:F3").Select
                  Selection.Copy
              Sheets("Signals").Select
                Range("Data").Cells(rowcount, 2).Select
                  Selection.PasteSpecial Paste:=xlValues
            End If
        Next rowcount
End Sub
End Sub
```

2-2 Matlab Code for Market Analysis

Function – Moving Average

```
function H = DMAC(eIndex,stocktype,speriod,lperiod,mxargus)

TRule=-1;


% Calculate Short & Long Averages
    for i=1:eIndex
      sAverage = subAverage(mxargus,stocktype,i,speriod);
      lAverage = subAverage(mxargus,stocktype,i,lperiod);
    end


% Apply Trading Rules
      if sAverage>lAverage
         TRule = 1;                    % Buy
      else
         TRule = 0;                    % Sell
      end
    TDecision = TRule;

% Return Trade Rule
    H = TDecision;


% Utilities
    function vAverage = subAverage(margus,stype,aIndex,period)
    sTotal = 0;
    lBound = aIndex - floor(period /2);
    if lBound<1
      lBound = 1;sTotal = sTotal + margus(lBound,stype);
    end
    uBound = aIndex + floor(period/2);
    if uBound>length(margus)
      uBound = length(margus); sTotal + margus(uBound,stype);
    end
      for j = lBound:uBound
         sTotal = sTotal + margus(j,stype);
      end
    vAverage=sTotal/period;
```

Function – ADX

```
function H=RSI(eIndex,nDays,ET,mxargus)
```

```
% Initialization
    eNo = eIndex;ETR=false;ETiR=false;RSt=0;vRSII=0;
    nInitial = eNo - nDays;TRule=-1;
    if nInitial<=1
      H = 0;
      return
    end
    nClose = 5;UClosetot = 0;DClosetot = 0;
    for i=nInitial:eNo
      UDCloset = mxargus(i,nClose) - mxargus(i-1,nClose);
      if UDCloset > 0
        UClosetot = (UClosetot*(nDays - 1) + UDCloset)/nDays;
      else
        DClosetot = (DClosetot*(nDays - 1) + UDCloset)/nDays;
      end
    end
      RSt=UClosetot/DClosetot;
      vRSII=100-(100/(1+RSt));


% Apply Trading Rules
    if vRSII>ET
      if not(ETR)                %RSI falls below ET and rises back
      TRule = 1;
      end
      ETR=true;
    else
      ETR=false;
    end
    if vRSII>(100-ET)
     ETiR=true; TRule = 0;       % Sell
    else
      if ETiR                    %RSI rises above (100-ET) and falls back
      TRule = 0;
      end
      ETiR=false;
      TRule = 1;                 % Buy
    end


% Return overall Profit
    H = TRule;
```

Function – Bollinger

```
    function [mav, uband, lband] = bolling(asset, samples, alpha, width)
```

```
% build weight vector
    i = (1:samples)';
    w = i.^alpha./sum(i.^alpha);

% build moving average vectors with for loops

    a = zeros(r-samples,1);
    b = a;
    for i = samples:r
        a(i-samples+1) = sum(asset(i-samples+1:i).*w);
        b(i-samples+1) = width * sum(std(asset(i-samples+1:i)).*w);
    end


    if nargout == 0
        ind = samples:r;
        h = plot(ind,asset(ind),ind,a,ind,a+b,ind,a-b);
        if get(0,'screendepth') > 1
            cls = get(gca,'colororder');
            set(h(1),'color',cls(1,:));
            set(h(2),'color',cls(2,:));
            set(h(3),'color',cls(3,:));
            set(h(4),'color',cls(3,:));
        end
    else
        mav = a;
        uband = a+b;
        lband = a-b;
    end
```

3-Apendix for optimization process

3-1 Excel Code for Optimization Analysis

Main Function – Optimization

```
Sub Optimize()
SolverOk SetCell:="$E$3", MaxMinVal:=2, ValueOf:="0", ByChange:="$C$6:$C$8"
   SolverSolve userFinish:=True
```

Function – Calculations

```
Sub Calculation()
```

-- Prompts inputs from the user --

```
   Range("C3").Select
   ActiveCell.FormulaR1C1
   Range("F3").Select
   ActiveCell.FormulaR1C1
   Range("C4").Select
   ActiveCell.FormulaR1C1
   Range("C5").Select
   ActiveCell.FormulaR1C1
   Range("D4").Select
   ActiveWindow.SmallScroll Down:=-3
   ActiveCell.FormulaR1C1
   Range("D5").Select
   ActiveCell.FormulaR1C1
   Range("E4").Select
   ActiveCell.FormulaR1C1
   Range("E5").Select
   ActiveCell.FormulaR1C1
   Range("E6").Select
   Sheets("Results").Select
   Range("C2").Select
```

-- Inserts inputs gathered from the user --

```
ActiveCell.FormulaR1C1 = "=Input!R[1]C[3]"
   Range("C3").Select
   Sheets("Input").Select
   Range("C6").Select
   ActiveCell.FormulaR1C1
   Range("D6").Select
   ActiveCell.FormulaR1C1
   Range("E6").Select
   ActiveCell.FormulaR1C1
```

```
Range("F6").Select
```

-- Calculation of Max Weight --

```
Sheets("Results").Select
ActiveCell.FormulaR1C1 = "=R[3]C*R[3]C[3]+R[4]C*R[4]C[3]+R[5]C*R[5]C[3]"
```

-- Returns Calculation --

```
Range("F6").Select
ActiveCell.FormulaR1C1 = "=AVERAGE(Input!R[1]C[-3]:R[10]C[-3])"
Range("F7").Select
ActiveCell.FormulaR1C1 = "=AVERAGE(Input!RC[-2]:R[9]C[-2])"
Range("F8").Select
ActiveCell.FormulaR1C1 = "=AVERAGE(Input!R[-1]C[-1]:R[8]C[-1])"
```

-- Standard Deviation Calculation --

```
Range("E3").Select
ActiveCell.FormulaR1C1 = _
   "=SQRT((R[3]C[2]*R[3]C[-2]^2)+(R[4]C[2]*R[4]C[-2]^2)+(R[5]C[2]*R[5]C[-
2]^2)+2*R[3]C[-2]*R[4]C[-2]*R[3]C[5]+2*R[3]C[-2]*R[5]C[-2]*R[3]C[6]+2*R[4]C[-
2]*R[5]C[-2]*R[4]C[6])"
```

-- Covariance Calculation--

```
Range("G6").Select
ActiveCell.FormulaR1C1 = "=VAR(Input!R[1]C[-4]:R[10]C[-4])"
Range("G7").Select
ActiveCell.FormulaR1C1 = "=VAR(Input!RC[-3]:R[9]C[-3])"
Range("G8").Select
ActiveCell.FormulaR1C1 = "=VAR(Input!R[-1]C[-2]:R[8]C[-2])"
Range("I6").Select
ActiveCell.FormulaR1C1 = "=VARP(Input!R[1]C[-6]:R[10]C[-6])"
Range("I7").Select
ActiveCell.FormulaR1C1 = "=R[-1]C[1]"
Range("J6").Select
ActiveCell.FormulaR1C1 = _
   "=COVAR(Input!R[1]C[-7]:R[10]C[-7],Input!R[1]C[-6]:R[10]C[-6])"
Range("K6").Select
ActiveCell.FormulaR1C1 = _
   "=COVAR(Input!R[1]C[-8]:R[10]C[-8],Input!R[1]C[-6]:R[10]C[-6])"
Range("J7").Select
ActiveCell.FormulaR1C1 = "=VARP(Input!RC[-6]:R[9]C[-6])"
Range("K7").Select
ActiveCell.FormulaR1C1 = _
   "=COVAR(Input!RC[-7]:R[9]C[-7],Input!RC[-6]:R[9]C[-6])"
Range("I8").Select
ActiveCell.FormulaR1C1 = "=R[-2]C[2]"
Range("J8").Select
ActiveCell.FormulaR1C1 = "=R[-1]C[1]"
Range("K8").Select
ActiveCell.FormulaR1C1 = "=VARP(Input!R[-1]C[-6]:R[8]C[-6])"
Range("H6").Select
ActiveCell.FormulaR1C1 = "=IF(ISBLANK(RC[-6]),"""",RC[-5]*Input!R3C3/RC[-6])"
Range("H7").Select
```

```
    ActiveCell.FormulaR1C1 = "=IF(ISBLANK(RC[-6]),"""",RC[-5]*Input!R3C3/RC[-6])"
    Range("H8").Select
    ActiveCell.FormulaR1C1 = "=IF(ISBLANK(RC[-6]),"""",RC[-5]*Input!R3C3/RC[-6])"
    Range("D10").Select
End Sub
```

4- Appendix for the verification of the optimization

4-1Matlab Simulation Codes for Efficient Frontier

```
function portfolio_eff_frontier(varargin)
if nargin == 1
% A set of tick prices was passed into the function - calculate returns
    x = varargin{1};

% Calculate returns
    r = x(2:end,:) ./ x(1:(end-1),:) - 1;
elseif nargin == 0
    daily_ret_lim = 0.05;
    r = 2*daily_ret_lim*rand(400,8) - daily_ret_lim;
end

% Calculate expected return and expected covariance
ExpReturn = mean(r);
ExpCovariance = cov(r);

% Convert daily return to annual
ExpReturn=ExpReturn*252;

% invoke the "portopt" function to produce the efficient frontier
npts = 100;
[PortRisk, PortReturn, PortWts] = portopt(ExpReturn,ExpCovariance,npts);

% Convert return to percentage
PortReturn = PortReturn*100;

% Plot Efficient Frontier
h_f = figure('units','norm','pos',[0.1 0.1 0.8 0.8]);
h_a = axes('units','norm','pos',[0.1 0.1 0.73 0.8]);
% Plot the curve, add a grid, and display labels
plot(h_a,PortRisk,PortReturn,'-b.');
grid(h_a,'on')
xlabel(h_a,'Risk (Standard Deviation)')
ylabel(h_a,'Expected Return')
title(h_a,'Mean-Variance-Efficient Frontier')
hold(h_a,'on');

end
```

References:

Papers and books:

1- Huaiqing Wang, John Mylopoulos, and Stephen Liao" Intelligent agents and financial risk monitoring systems"
2- Vikas Agarwal , Narayan Y. Naik "Introduction to Hedge Funds"
3- William F. Sharpe," Expected Utility Asset Allocation"
4- Jim Smith, "Optimum portfolio analysis"
5- Mark Austin , "Introduction to the system engineering"
6- Richard A. Ferri, " All about asset allocation"
7- N. V. Batalova, V. Maroussov &F. G. Viens," Selection of an optimal portfolio with stochastic volatility and discrete observations"
8- Sengupta, Chandan. Financial Modeling. New Jersey: John Wiley & Sons, Inc., 2004.
9- Jackson, Mary, and Mike Staunton. Advanced Modeling in Finance. New Jersey: John Wiley & Sons, Inc., 2001.
10- Winston, Wayne L. Financial Models Using Simulation and Optimization. 2nd ed. New York: Palisade Corporation, 2001.
11- Schach, Stephen R. Introduction to Object-Oriented Analysis and Design. New York: McGraw Hill, 2004.

Websites:

1-http://advisor.investopedia.com/freereport/default.asp
2-http://www.investopedia.com/terms/p/portfolio.asp
3-http://www.mysavingsatwork.com/taxexempt/assets/AA_Planner.pdf
4-http://www.schwab.com/public/schwab/research_strategies/portfolio_management/
5-http://en.wikipedia.org/wiki/Modern_portfolio_theory
6-http://miningstocks.com/max
7-http://www.mathworks.com/access/helpdesk/help/toolbox/modelsim/index.html?/
8-http://accounting.knowledgestorm.com/ksaccounting/search/tabkeyword/research/StockPortfolio+Management+Solutions/1/index.jsp
9-http://www.schwab.com/public/schwab/research_strategies/portfolio_management/index.html?src=mxf
10-http://finance.yahoo.com/q
11-Technical Indicators Explained | IQ Chart
12-http://www.mathworks.com/index.html?ref=pt
13-http://transcripts.fxstreet.com/2005/10/date_october_4_.html
14-http://gold.globeinvestor.com/public/help/flat/help_charts_ta_oscillator.html
15-http://www.crystalball.com/spotlight/printer_friendly/spotlight20_print.html