***Option Pricer System***

***Requirement Document***

*Sameer Patel, Bhargavi Reddy, Angela Gu, Luna Yang*  
*February, 02, 2015*  
*Version 1*

**1. Introduction**

**1.1 Purpose of this document**

This document describes the requirement specifications for the Option Pricer System of CMUA. The project is sponsored by the financial division, and is the intended audience of this document.

**1.2 Scope of this document**

This document is divided into seven sections.  It provides an introduction and overview of the Option Pricer System, a general description of the system and its users, functional requirements in detail, interface requirements of how the software interfaces with users, performance requirements, design constraints, and other non-functional requirements.

The requirements in this document were determined through interviews with the staff of CMUA and requests from the traders.

**1.3 Overview**

The Option Pricer System is a system application that calculates the price using different algorithms, displays a consistent result quickly, compares the results with each other and analyses the result more accurately. Unlike our previous system that produces inconsistent results with different algorithms and lets traders take advantage of the inaccuracy, the OPS will be consistent, more accurate and more efficient.

**1.4 Business Context**

CMUA is a financial firm that provides pricing options for the traders. The mission, goal and organizational structure of the company is as followed:

Mission – To be the pioneer in the financial consulting industry and give customers a perfect trading experience.

Goal - To provide customers with improved, efficient and accurate stock price calculation and thereby effective trading consultation.

CMUA is headed by a President, assisted by a Vice President and an Executive Staff composed of the associate directors. CMUA has six divisions and the software team is responsible for the development of the new Option Pricer System.

**2. General Description**

**2.1 Product Functions**

The unique functions of the new Option Pricer System is to calculate option prices over different regions and generate volatility graphs, allow the desk quants to select from the possible algorithms to calculate an option price and compare their results.

For calculating the different option prices, the OPS requires users to provide certain parameters:

* Current stock price, e.g. $100.00
* Duration, e.g. 3 month
* Expend growth rate, e.g. 0.04
* Stock price volatility, e.g. 0.2
* Risk free interest rate, e.g. 1%
* Option side, e.g. call or put
* Type of option, e.g. American, European, Asian

**2.2 Similar System Information**

Currently, CMUA uses multiple set of programs across various trading desks to calculate option prices which is confusing and not reliable. The traders need a new Option Pricer System to avoid the inconsistency which arises through using different programs.

The new Option Pricer System will be a consistent and integrated program which retrieves stock data from the Stock Exchange Service.

Some competitors of the company use similar software, but their system are not stable and highly prone to errors. The OPS is first of its kind available in the industry.

**2.3 User Characteristics**

Option Pricer System will be used by all traders of CMUA. Traders need to use the OPS to calculate the price using different pre-defined algorithms with different option types, add new algorithms and compare the results with each other.

All traders will have access to the application and have a computer capable of running the software.

**2.4 User Problem Statement**

There is a possibility that the trader will not use the OPS effectively and would go back to their previous system which would lead to a significant loss of investment. The OPS lets traders to add new algorithms to calculate option prices. The new input algorithm may need new variables. Users may not understand the working technicalities of the pre-defined algorithms. Traders need essential descriptions for using the OPS.

**2.5 User Objectives**

The Option Pricer System will support several algorithms and types of option prices and let traders compare the results of different algorithms.

The Option Pricer System will display graphs of volatility smiles.

The Option Pricer System will allow traders to add their own algorithm into the system.

Traders should be able to run the software. The result will be returned within 0.5 sec with 1% error margin.

**2.6 General Constraints**

The Option Pricer System will run on Windows and iOS platform. Traders of CMUA can use the Option Pricer System to calculate the price using different algorithms.

The time from when query is submitted to the result is generated should be in 0.5 seconds with 1% error margin. If the time estimated exceeds the limits, the system will notify the traders about the time.

**3. Functional Requirements**

**3.1 Calculate stock option price**

**3.1.1 Description**

This functionality is part of the integrated Option Pricer System to help traders calculate and compare price options using several algorithms. A user interface is designed such that the trader selects the region of Stock option and the algorithm from the list of algorithms (Binomial trees/Simulation/Numerical Integration). The user then enters a set of parameters relevant to the selected algorithm in order to calculate the stock option price. The system runs the algorithm and calculates the stock option price.

**3.1.2 Criticality**

This is the primary purpose of the Option Pricer software, therefore it is highly critical to ensure the system’s success.

**3.1.3 Technical issues**

The user interface must be created to ensure the input parameters are accurately entered by the user to calculate the stock option price. The input fields should only allow a range of acceptable values and signal an error if the inputs do not meet the required data types.

**3.1.4 Risks**

There is a risk of incorrect calculation of option price in case the input parameters are incorrectly entered by the user. Logical errors or runtime errors might occur in the calculation.

**3.1.5 Dependencies with other requirements**

This functionality fully depends on the input parameters provided by the user/trader:

* Current stock price, e.g. $100.00
* Duration, e.g. 3 month
* Expend growth rate, e.g. 0.04
* Stock price volatility, e.g. 0.2
* Risk free interest rate, e.g. 1%
* Option side, e.g. call or put
* Type of option, e.g. American, European, Asian

It also depends on the algorithm selected by the user namely Binomial trees, Simulation or Numerical Integration which will determine the output.

**3.2** **Generate Volatility Smile (VS):**

**3.2.1 Description:**

Once, the input values are entered, the data points for resulting graph (volatility smiles) are calculated. It will also save input data along with results save the type of stock option, and other entered parameters like the algorithms delta range etcetera.

**3.2.2 Criticality**

It is critical that volatility smiles be generated which would be presented in a graphical format to the user. This will help users compare different algorithm which is an important task in the OPS.

**3.2.3 Technical Issues**

The calculation within the range should be completed in 0.5 seconds. The range of the graph and readability might be affected by data anomalies.

**3.2.4 Risks**

No risks are involved in generating the results.

**3.2.5 Dependencies with other requirements**

The VS will depend on the valid parameters from the user. This has been addressed in functional requirement 3.1. This requirement also interacts with the Help Function in the case the trader wants more information or experiences an error in generating volatility smiles.

**3.3 Display result**

**3.3.1 Description**

The result of the executed algorithm of calculating stock option prices should be displayed to the user along with the Volatility smile curve in less than 0.5 sec. If the algorithm successfully runs, the results are displayed on the screen along with the input parameter values. Users are then given an option to make a new selection.

**3.3.2 Criticality**

Displaying the results to the trader is integral to the system, hence this is a critical functionality.

**3.3.3 Technical issues**

A GUI interface needs to be designed to appropriately display the results of the selection. The original values of parameters entered to calculate the price must be retained by the system to allow users to modify the selection or make a new selection. The result should be within an error margin of 1% and should be displayed within 0.5 sec.

**3.3.4 Risks**

There is no risk involved in displaying the results.

**3.3.5 Dependencies with other requirements**

This functionality depends on the successful execution of the algorithm in calculating the stock price and volatility smile.

**3.4 Add custom algorithm**

**3.4.1 Description**

This functionality of the Option Pricer system enables the user to add their own custom algorithms into the system to calculate stock option prices. The user interface is designed so that the user can opt to add an algorithm. The user enters the algorithm and additional variables if any. The program checks for syntactical errors. If the algorithm is syntactically correct, it is added to the user’s list of existing algorithms to choose from. If it is syntactically incorrect, it displays an error message signaling the occurrence of an error.

**3.4.2 Criticality**

As this functionality is not required for the Option Pricing System to work, it is not a critical requirement. However, it offers flexibility to the user to customize the software according to his needs.

**3.4.3 Technical issues**

GUI in the user interface must be designed so that the user can opt to add an algorithm. The interface for the algorithm to be entered has to be designed. The system should also be configured to verify the algorithm for syntactical errors and add the algorithm to the list of options available for calculating option prices. Verifying if the custom algorithm is correct and accurate is another technical issue.

**3.4.4 Risks**

The accuracy and efficiency of the custom algorithm is not tested. The GUI interface to allow a user to create a custom algorithm also poses a risk. It is also difficult to predict the run time or processing time of the algorithm.

**3.4.5 Dependencies with other requirements**

This functionality depends on the ability of the system to accurately run the custom algorithm once the user adds it to calculate the Option price. It also depends on whether the algorithm is syntactically correct.

**3.5** **Provide online help and tutorial**

**3.5.1** **Description**

Extensive help will be provided to the traders. The Help section will cover topics explaining the functionality of the system: calculating stock option pricing, generating volatility smiles, adding an algorithm, and a troubleshooting section. The Help topics will be indexed to provide context sensitive help from error messages received in other parts of the application or by clicking the Help button on a screen. There will be tutorial video that can be completed in under 5 minutes each will cover topics such as, Getting Started, How to Create a Query, and Using Advanced Features of the software.

**3.5.2 Criticality**

This is not a critical function. As the traders are switching from legacy system to a new system, it is important to have a easy to understand help section. It is also important to increase usability of the system.

**3.5.3 Technical Issues**

Every GUI element in the user interface should be linked to specific help sections to provide immediate help regarding the section of interest.

**3.5.4 Cost and schedule**

Creation of tutorialwill be additional activity with minimum cost requirement. The documentation will be started with development and constantly updated throughout the software development life cycle.

**3.5.5 Risks**

There is no risk associated with this requirement.

**3.5.6 Dependencies with other requirements**

This requirement interacts with all other aspects of the system to provide assistance to the user.

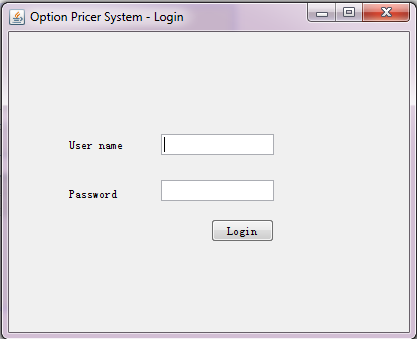
**4. Interface Requirements**

This section describes how the software interfaces with users for input or output.

**4.1 User Interface**

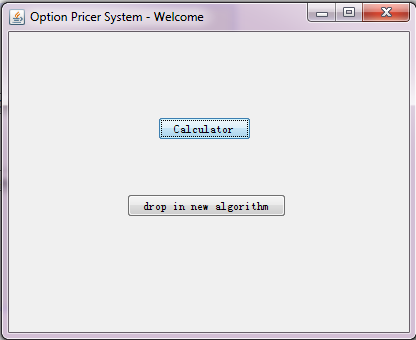
This is the prototype of the system, and is only for visualization purposes. It is not a finished product.

**4.1.1 Login interface:** Users need to enter user name and password to login the system.



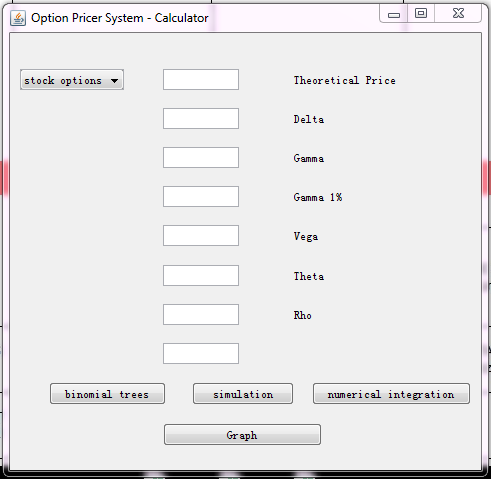
**4.1.2 Welcome interface:**

In the welcome interface, users can choose to start the calculation or to drop in new algorithm.



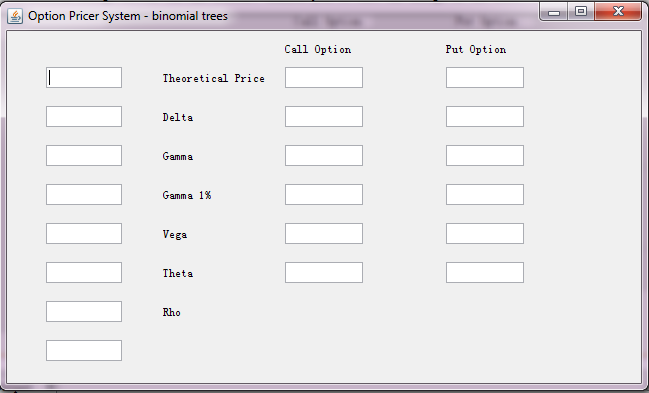
**4.1.3 Calculator interface:**

1. Choose stock option in this interface like call or put with type American or European or Asian.
2. Put in data in the required areas
3. Choose the algorithm type / graph and a new window will appear to show the result.



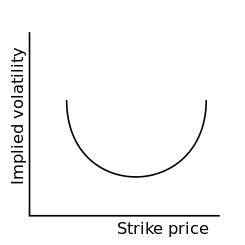
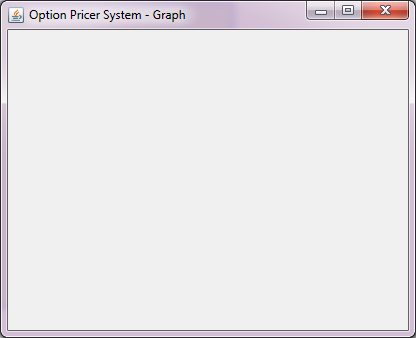
**4.1.4 Result interface**

This window shows output corresponding to the selected algorithm and input parameters.



**4.1.5 Graph interface**

This window shows up when user clicks the graph button.



**4.1.6 Diagnostics or ROM**

The tools provided in the IDE (Netbeans) of the programming language will be used for diagnostics.

**4.2 Hardware Interfaces**

The system does not interact with hardware interfaces.

**5. Performance Requirements**

The response time will be within 0.5 seconds with 1% error margin.

Query response time will vary depending on the type and the complexity of the query, and the speed of CPU of the operating computer. This maximum speed is much faster than the calculation speed offered by the original software and by competitor companies. If the query is expected to exceed the maximum response time, the user will be informed.

The error rate is much lower than the error rate of the previous software. The algorithms embedded in the software are very reliable.

**6. Design Constraints**

**6.1 Standards Compliance**

The system will be designed with industry standard and in compliance with the CMUA’s design standards. GUI interface will be designed according to windowing environment and standards.

**6.2 Hardware Limitations**

The system does not interact with hardware interfaces.

**6.3 Security Constraints**

The system will run on multiple computers. With their unique username and password, users are able to log in the system.

Data used for calculation will be recorded only when the user updates it after every calculation.

**7. Other Non-Functional Attributes**

**7.1 Security**

Confidentiality requirements are very important to the system.

Users need their unique username and password to log in the system.

The system does not have a functionality to store data.

**7.2 Reliability**

The system will be designed to restart with the former state when system failure occurs.

**7.3 Maintainability**

The system will allow users to add new algorithms for calculation. It will also allow programmers to extend it to other types of products, like futures, swaptions and so on.

**7.4 Portability**

The system shall function identically on Windows and iOS.

**7.5 Extensibility**

It is extendable to other types of products, like futures, swaptions and so on.

**7.6 Usability**

The system is designed to have very simple and easily understood interface. It will be very easy to navigate and to run queries. With minimum training, the user will be able to operate the system efficiently.

**7.7 Responsiveness**

Depending upon the complexity of the query, the response times shall not exceed the limit of 0.5 seconds. The company and the design staff will mutually agree with the calculation time limits for the queries.

The Complexity levels will be determined by the size of data and the algorithms the user choose. If the query is expected to exceed the maximum response time, the user will be informed. Users will be given the option to cancel, modify, or run the query.

**7.8 Accuracy**

The result of the calculation will have a maximum 1% error margin.