Software Requirements Specification

for

ThisTrading

**Version 0.0.1**

**Prepared by Bravo Team   
Ashutosh**

**Karalyn**

**Shalki**

**Stacey**

**Citi Tech MA**

**Date**

# Overall Description

## Product Functions

This Workflow application is a stock system. The users will be able to use the web page by the system that is in place. The intended users are Administrators, Managers, and Analysts who intend to perform specific tasks.

By using this system the users will:

1. Log in to the system,
2. Administrator will have the ability to create users.
3. This will send the user to the appropriate location.
4. When in the location, the user will have the authority to check, modify, verify or complete workflow tasks according to the rights that are allocated.
5. The workflow will have stages that are modelled as Queues, to be follow in order to amend the files.
6. Unknown stock, missing data and invalid dates will be present
7. The number of items in each workflow will be able for display.
8. Overdue workflow items and duration will be highlighted.
9. Selected workflow items to be compact and listed by age.
10. Displaying of all details of a selected workflow item (which include: read-only files, editable files, highlighting files that have had verification tests failed and an audit trail)

## Technologies

* + 1. WorkFlow will be a tree tier project.
    2. Front, middle and back end.
    3. The technology being used is HTML, CSS, JavaScript, Json, Ajax, C# and SQL.
    4. The user will have experience HTML, CSS, JavaScript in the front end

## Design Goals

* + 1. The system will be a simple design interface.
    2. The system will allow the Administrator to register a user and make decisions and audit information.
    3. The system will allow a workflow Queue to be verified, in progress, await authorisation, and to be complete in unknown stock and missing data.
    4. The system will allow the a process of Open, in progress and completed in invalid date
    5. accept an item and allow one of the users(analyst to sent comments)

# Functional Requirements

***2.1 Algorithmic Trading***

*2.1.1 The application must contain at least three algorithms, i.e. Two Moving Averages,   
 Bollinger Bands and Price Breakout.*

*2.1.2 The application will allow the user to specify the stock to trade on and the algorithm to   
 be executed on the stock.*

*2.1.3 The algorithm will run indefinitely upon execution unless specified otherwise.*

*2.1.3.1 The algorithm will stop its execution upon a given exit condition is met.*

*2.1.3.1.1 The application will allow the user to input a stop loss condition.*

*2.1.3.1.2 The application will allow the user to input a take profit condition.*

*2.1.3.2 The application will allow the user to manually stop an execution.*

***2.2 Transactions***

*2.2.1 The application will be able to send a buy or a sell trade order down to the QuickFixJ   
 exchange through an Order Broker.*

*2.2.2 The application will be able to handle messages from the OrderBroker.*

*2.2.2.1 The application will be able to handle transaction success messages from   
 the Order Broker.*

*2.2.2.2 The application will be able to handle transaction error messages from the   
 Order Broker.*

*2.2.2.2.1 The application will be able to handle a invalid transaction error   
 from the Order Broker.*

*2.2.2.2.1.1 The application will handle invalid transactions such as when trader wants to trade a stock over its limit.*

*2.2.2.2.2 The application will be able to handle a rejected transaction error   
 from the Order Broker.*

***2.3 Login User Interface***

*2.3.1. The application will allow the user to log in to the system through the login interface.*

*2.3.1.1 The user must be a registered user in the system to be able to access the system.*

*2.3.1.2 The user must insert his username and password to access the system.*

*2.3.1.2.1 The application will return an error message upon if the username and password combination is incorrect.*

*2.3.1.2.2 The application will return an error message upon invalid input.*

*2.3.1.2.2.1 The user must input a non empty field in his username field.*

*2.3.1.2.2.2 The user must input a non empty field in his password   
field.*

*2.3.1.3 The application will authenticate the user and redirect the user to the   
 dashboard interface.*

*2.3.2 The application will allow the user to register himself through the login interface.*

*2.3.2.1 The user must input the username, the password and the confirm password   
 to register himself.*

*2.3.2.2 The application will return an error message upon invalid input.*

*2.3.2.2.2.1 The user must input a non empty field in his username field.*

*2.3.2.2.2.2 The user must input a non empty field in his password field.*

*2.3.2.2.2.3 The user must input the same fields in his password and confirm password   
 field.*

*2.3.2.2.2.4 The user must input a username that is not taken.*

***2.4 Dashboard User Interface***

2.4.1 The application will show a list of available stocks in the Singapore Exchange.

2.4.1.1 The application will allow the user to filter through the list of stocks by the stock symbol.

2.4.1.1 The application will allow the user to filter through the list of stocks by the stock name.

2.4.2 The application will show the information regarding selected stocks on the dashboard interface.

2.4.2.1 The application will allow the user to select stocks to be shown on the dashboard interface.

2.4.2.2 Multiple stocks can be shown on the same dashboard interface at the same time.

2.4.2.3 The user can rearrange the order of appearance of the stocks on the dashboard interface.

2.4.2.3 The application will show information of the stocks in the form of graphs.

2.4.2.3.1 The application will allow the user to choose the time range of the graphs

2.4.2.3.1.1 The application will allow the user to choose time ranges of 1 minute, 15 minutes and 1 hour.

2.4.2.4 The application will allow the user to manually refresh the information of each graph

2.4.3 The application must allow the user to execute an algorithm on a particular stock.

2.4.3.1 The application will request the amount of money to be input for each stock.

2.4.3.1.1 The application will return error upon invalid input.

2.4.3.1.1.1 The input amount over its limit or the input amount is a negative value.

2.4.3.2 The application will request the stop-loss and the take-profit parameters for each algorithm.

2.4.3.2.1 The application will return error message upon invalid input.

2.4.3.2.1.1 The input amount is a negative or an empty value.

2.4.4 The application must allow the user to manually halt the execution of an algorithm through the dashboard interface.

2.4.4.1 The application will allow the user to select halt the algorithm based on the selected stock.

2.4.4 The application will supply the user with real-time alerts.

2.4.4.1 The application will notify the user through the interface when the selected algorithm’s successful stock transactions through *QuickFixJ* .

2.4.4.2 The application will notify the user through the interface upon his selected algorithm’s failed stock transactions through *QuickFixJ* .

2.4.4.3 The application will notify the user through the interface when his selected algorithm’s issues a stock transactions through *QuickFixJ* .

**2.5 History User Interface**

2.5.1 The application must provide the algorithm’s trading history on the history user interface.

2.5.1.1 The application will show a history of an algorithm’s successful transactions.

2.5.1.2 The application will show a history of an algorithm’s failed transactions.

2.5.1.3 The application will show a history of an algorithm’s current pending transactions.

***2.6 User Settings***

2.6.1 The system will allow user to change the password.

2.6.2 The system will return an error upon invalid input.

2.6.2.1 The system will return an error if the user inputs a blank password.

# Non-Functional Requirements

*3.1. Availability*

Thesystem must be available on a local network via a web page and should be able to be accessed by specific users. The users will be Administrator, Manager and Analyst. Only tasks that is assigned to each user will be available to them.

*3.2. Security*

The system must be secure and allow only registered authorized users to access the system interface that they have been assigned to by the Administrator.

*3.3. Stability*

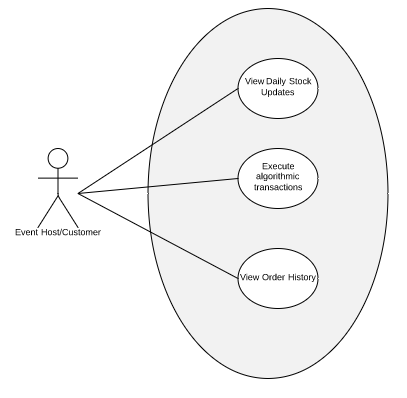
The system must be stable and experience minimal to no downtime and provide continuous service to users. This will be done a clear and precise manner.

*3.4. Accessibility and concurrency*

The system should be accessible to multiple concurrent users, and the actions of one user should not affect what a different user can do. Some users should have more accessibility than others.

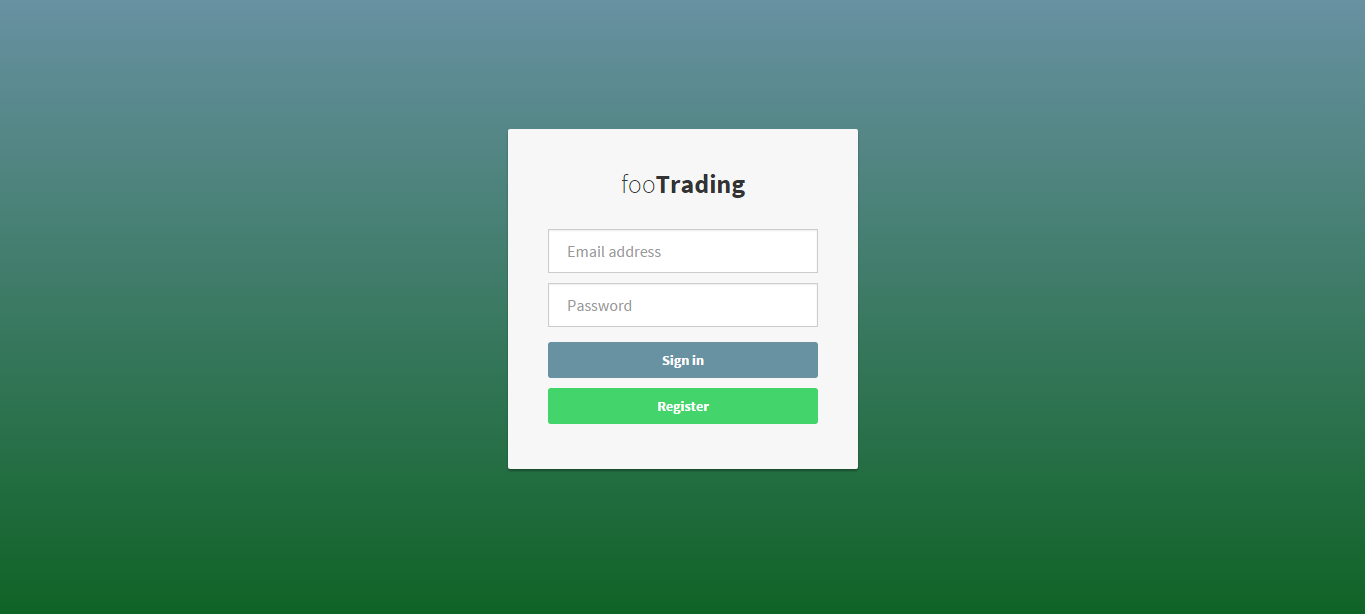
# Appendix

# Use Case Diagram



# User Interface Mockup

*b1. Login Interface*



b2. Dashboard interface

