

# Requirements Analysis Report

## **Group 7**

Harry Stevenson  
Stephen Lynn  
David Todd  
Liam Joy  
Thomas Gak Deluen

February 1, 2017

## 1 Abstract

This document is the use of the stakeholders (Deutsche Bank) and the assessor (Professor Stephen Jarvis), along with the software developers, it will outline the requirements of the software to be developed for these stakeholders, for discussion in a meeting on Friday 10th February.

## 2 Introduction

The task set is to produce software to help the stakeholders identify and flag anomalous trading patterns on the London Stock Exchange for stocks within the FTSE 100. This is to provide an early warning system to alert the bank to rapid market changes that would be too fast to spot manually. Without this, the stakeholders would be left vulnerable and less aware to the events causing major fluctuations within the stock market and potentially catch rogue traders, looking to exploit the market using specified, known techniques. The identification of such events must be done in a live data environment and the software must be capable of coping with the fast-moving nature of the dataset required, thus human intervention should only be required as necessary.

## 3 Glossary

Stakeholders - Deutsche Bank

## 4 User Requirements

### 4.1 Functional Requirements

The Software must be able to learn the characteristics of the stocks in the FTSE 100 using a machine learning algorithm and send an alert to notify the Business Analysts of the stakeholders when one of the following trading patterns is detected. This will then allow the user to obtain the relevant data to perform the root cause analysis, through the display of graphs or other relevant data display methods. These patterns can be categorised as follows:

- **Identify Fat finger errors** - human error entering data, this may include incorrectly inputting the volume or price of trades. The algorithm must be able to detect when abnormalities occur in data inputted, i.e. volume or price values are larger than average by a factor of ten.

- **Pump & Dump** - illegal buying of large share volumes to artificially inflate the price, by encouraging others to follow the trend. Shares can then be sold at a higher price, yielding a large profit for the rogue trader. The software should check when a single or multiple trader IDs buys a large volume of shares and sells them shortly afterwards for the new higher price.
- **Bear Raid** - opposite to pump and dump, a trader or group of traders may short a high volume of stocks, thus manipulating the price of the stocks, sending the price down, they will then buy back these shares at the lower price hence making a profit. In a similar manner to pump & dump, an algorithm must check that traders are not selling large volumes of shares to then buy them back in a short amount of time.
- **Volume Spikes** - when compared to the average trend for the volume of trades of a share/group of shares, if there is an unusually high volume of these stocks being traded at either once or spread out over a certain period. This could indicate that methods such as pump & dump or bear raids are taking place and thus should be monitored. The software should compare the volume of each trade to the average volume for this stock as determined by the machine learning algorithm which will be dependent on the fluctuation of this stock.

In doing this the software must be able to analyse both a live feed for up to date data, in addition to a static csv file to analyse historical data in order to make comparisons. This historical data will also be available to the user for them to drill down into the data which caused the alert, this data should date back to the previous two days such that meaningful short term predictions and comparisons can be made.

The system should be able to cope with the connection and reconnection of the server from which it retrieves the data. Therefore, when the connection resets itself at 12AM, the software should be able to disconnect safely, retaining all data and perform the update of historical data and then be able to reconnect to the host on port 80 before 1AM when the server resumes publishing data.

The system does not require to be able to connect to any external data feeds or support multiple users, security of the system along with the Deutsche Bank branding should not be part of the implementation.

In addition to these mandatory, primary requirements, the software may also include the following features: Automatically organise possible trading anomalies in to the categories described above, this could then improve the detection and tagging of anomalies in the future, through the machine learning algorithm. Cope with the possible manipulation of whole sectors of the market or selected groups, this would allow users to browse the data provided by market sector or by group of individual traders through a search

function or tick boxes. Use the historical data collected and archived to predict the short-term future price and volume trends, again aiding the machine learning algorithm in deciding on thresholds beyond which trades are considered anomalous. The user could be allowed to assist and adapt the machine learning algorithm, this will mean that it can be tailored to a selected level of severity; however, it must be noted that the users will not be statistical or machine learning experts so these methods for adaptation must be implemented in a simple, intuitive way.

## 4.2 Non-Functional Requirements

In addition to the functional requirements of the system, non-functional requirements describing how the system works must also be considered for it to be effective in the workplace.

The system should always remain responsive whilst being able to process the large datasets required, thus it should be the case that delay between a rogue trading pattern occurring and being detected and displayed is at a minimum such that there is 5 seconds or less between the trade happening and the user being alerted to this.

The systems frontend will ensure a simple, effective display of the current system status, suggesting if action is needed to be taken or not. This has been chosen due to the many streams of data already in possession of the analysts utilising this software, thus it is not required to replicate all the data received, only the data of interest for the investigation of anomalous trading patterns. This allows users of a basic understanding to appreciate and recognise in what situations action should be taken, but also allowing the more advanced users to access the data required to respond to such patterns successfully.

A consistent, strong user interface is important to the design of the system, similar themes should be used throughout and the display of data should be done in an appropriate way such that it is easy to extract the data required to understand the situation, through the use of graphs and tables for each stock as required such that data visualisation is not excessive.

If the connection to the live data feed drops at any time during publishing of data (unless between 12am and 1am) then the system should be able to reconnect seamlessly to reduce the loss of data. The system should therefore keep checking the connection regularly and as soon as it detects that the connection has disconnected, then immediate reconnection should occur.

Before going live, the system will undergo rigorous testing (as laid out in the planning and design document) to ensure its smooth running and effectiveness, this will complement such unit tests that will be made throughout the design and implementation process.

## 5 Group Management

Through a democratic vote taken from within the group discussions, it has been decided that Thomas will take on the role of project manager, due to his knowledge of the task in hand and the experience he has had from similar projects in the past. He shall work alongside Stephen to create the frontend of the application, whilst still overseeing the work of Harry David and Liam on the frontend. The whole group will meet at least once a week in the already discussed available timeslots, although additional meetings may occur will necessary group members as required. All major decisions shall be taken as a group vote, with the project managers decision counting as final in the case of any tied votes, however, all decisions involving the frontend or backend team only, will be taken from within the relevant team, including the project manager, such that it allows consistency throughout the application. The planning and design of the application will be done in the teams prescribed above, with both teams informing each other throughout the process to ensure the feasibility and consistency of the application.