Crypto Alert System

Software Requirements Specification

**Prepared by**

Marcus Nardizzi 100327114  
Josiah Tolentino 100304547  
Avery Gibson 100330734  
Ceston Carino 100330308  
Anmol Singh Bains 100332252

**20th January, 2018**

**Table of Contents** [1](#_1fob9te)

**1.** **Introduction** 2

[1.1 Purpose 2](#_1fob9te)

[1.2 Scope](#_3znysh7) 2

1.3 Definitions, acronyms, and abbreviations 2

1.4 Overview 2

[**2.** **Overall Description**](#_1t3h5sf) 3

[2.1 Product Perspective](#_4d34og8) 3

[2.2 Product Functions](#_2s8eyo1) 3

[2.3 User Characteristics](#_17dp8vu) 4

[2.4 Constraints](#_3rdcrjn) 4

[2.5 Assumptions and Dependencies](#_26in1rg) 5

[2.6 Apportioning of requirements](#_lnxbz9) 5

**3.** **Specific requirements** 6

[3.1 External Interfaces](#_1ksv4uv) 6

[3.2 Functional requirements](#_44sinio) 6

3.2.1 Display Graph 6

3.2.2 Choose Cryptocurrency 6

3.2.3 Set First Trading Rule 7

3.2.4 Set Second Trading Rule 7

3.2.5 Set Third Trading Rule 7

3.2.6 Show Alert 7

[3.3 Non-functional requirements](#_2jxsxqh) 7

3.3.1 Performance Requirements 7

[3.3.2 Design Constraints](#_3j2qqm3) 7

[3.3.](#_3j2qqm3)3 Standards Compliance 7

3.3.4 Reliability 7

3.3.5 Availability 8

3.3.6 Usability 8

3.3.7 Logical Database Requirements 8

# Introduction

## Purpose

The purpose of this document is to provide the customer with a detailed description for the ‘Crypto Alert System’. The intended users of this application are those who are interested in cryptocurrency trading. It will cover the software requirements and illustrate how it will function internally and with external systems. It will also explain what the system will not do and its constraints.

## Scope

The ‘Crypto Alert System’ notifies the user when trading rules have been met. The buying and selling alert rules will be set by the user. The system will analyze the data constantly so the user does not need to monitor the activity. The application will not be doing any trading, but the user will be alerted when to do so from the rules provided.

There will be many cryptocurrencies that the user can select from, which will be displayed on a graph. The system will be using an API to get live data and will update frequently. This process requires an active internet connection.

## Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| API | Application programming interface |
| User | Someone who uses the ‘Crypto Alert System’ |
| Cryptocurrency | A digital currency |
| RAM | Random Access Memory |

## Overview

This SRS will present an outline of what the application will do, and the parts that make up the application.

# Overall Description

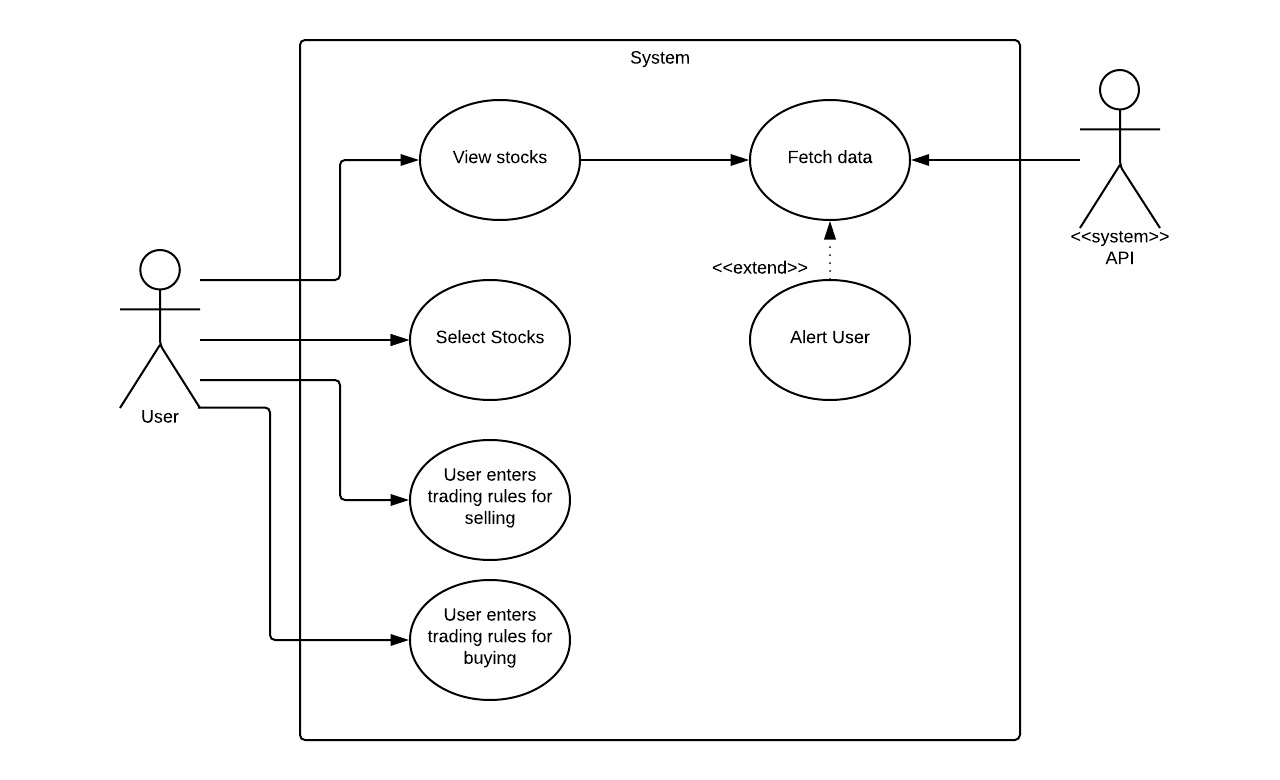
## Product Perspective

The purpose of this application is to make it easier for people to get into cryptocurrency trading. Cryptocurrency trading is extremely difficult for beginners as companies are emerging and shutting down every day and the changes are more rapid than any other industry.

People who have the resources, but do not have the time to keep a tab on the progress of their favored currencies or want to start trading, but do not have enough knowledge and want to make safe trades. The application will allow users to pick the currencies of their choice, and be alerted with the criteria they provide. We will provide the top 5 most reliable cryptocurrencies.

## Product Functions

The user will be able to select multiple crypto currencies to be displayed. The program will receive live data on the selected cryptocurrencies, by using an API, and be shown on a graph. The user may set buying and selling alerts by providing rules when the system should prompt the user. The system will periodically get live data and show on the graph. If the new changes meet the requirements set by the user, an alert will be shown.



## Use-Case Diagram for Crypto Alert System

User:  
 • Client-end actor who will be the one using the application.  
 • The user can view stocks provided the API.  
 • Out of the stocks provided, the user will select the currencies they   
 want to monitor.  
 • Once selected, the user will enter trading rules for selling and   
 buying.  
 • The rules can be viewed on 3.1 on external interfaces.  
 • If the trading rules are met, the system will alert the user based on   
 the updated data.  
  
Fetch Data:  
 • In order to get updates from the API, the user’s machine will have  
 to allow the API to send information to the machine.  
  
API:  
 • Application programming interface.  
 • This will be the main source where the system will get its  
 information from.

## User Characteristics

The expected users for this application are those people who are interested in getting into cryptocurrency trading, but do not have the time to follow each event closely or the knowledge, but still want to trade cryptocurrency.

## Constraints

Due to the nature of the programming language being used, the application will only run on windows based platforms. Moreover, since this is a desktop based application, it needs to always be active in the background in order for it to perform its tasks.

Since the application will typically used for long periods of time, it will use use considerable resources, and will require a machine that constantly receives power, and is energy efficient.

Even though there will be a cut off of amount of data the application will store, the data will be stored in the RAM. The user’s machine will require sufficient RAM to store data over long periods of time.

The application gathers the data it requires from online APIs, and this requires a steady internet connection.

The firewall will also need to be configured in order to allow the application to repeatedly make requests to the API.

The API we are using does not update its live data often.

## Assumptions and Dependencies

We are assuming that the user’s machine is always on, connected to the internet, and running our application; or else the application will not receive any data from the API.

The user has the latest version of .NET framework.

As the data is retrieved via an API, we assume the external API system is currently operational and data is obtainable.

## Apportioning of requirements

In the future we hope to implement a more dynamic graph for the representation of the data, and get more accurate data.

In addition, we would like to see an increase in support for more currencies and trading rules.

# Specific requirements

## External Interfaces

• The UI will have 5 tabs.  
• The main tab will show the historical data graph for the currencies, and will allow the user to pick the currencies.  
• The user will be allowed to choose up to 3 different cryptocurrencies.  
• The rest of the tabs will each display the graph for a specific crypto that the user has chosen and the progress on the trading rules, the trading rules will also be set here.

3 trading rules:  
 1. If the currency is above or below a certain point.  
 2. The currency falls or rises by an amount specified by the user.  
 3. The currency shows a continuous rising/falling trend.

## Functional requirements

### Display Graph

Display a graph using the data retrieved from the API of the cryptocurrencies that the user selected.

### Choose cryptocurrency

Let the user select up to 3 cryptocurrencies.

### Set first trading rule

The price rises or drops to a certain point.

### Set second trading rule

The price rises or drops by a specified amount.

### Set third trading rule

The currency has been rising or following.

### Show alert

Show an alert when one of the trading rules is met.

## Non-functional requirements

### Performance Requirements

The data must be fetched as often as the data from the API updates. This will be every 15 seconds.

The trading rules will be analyzed instantaneously when the incoming data is received.

Alerts must be calculated and displayed in less than a second.

The number of simultaneous terminals are innumerable since each application will be independent on each other and only depend on the API.

Computer RAM availability should be at least 1GB.

### Design Constraints

The Crypto Alert System can only run on an environment using the latest .NET framework.

There can only be one user on the app at the moment, there is no provision for multiple users.

### Standards Compliance

There will be consistency with variables and alerts within the system and the coding within.

The User Interface shall be uniform with its the design making the application visually consistent.

### Reliability

The system will requires a reliable and consistent internet connection.

The application depends on the API to consistently update its information.

If the API fails the application will be unable to update.

May not be compatible with older versions of .NET.

### Availability

Users will be able to have instant access to the latest cryptocurrency trends and alert system.

The alert system will activate when the target statement is met.

### Usability

The system should work without any training. The procedure should be simple enough to follow as they use for the first time. Helpful text will be used to help the trader make safe trades.

### Logical database requirements

Since our application is using an API to get real-time data from the trading markets, we do not need to host the data on the user’s device, hence we do not need a database. Although, the program will be keeping an array of all the data that it fetches from the API, in order to create the graphs. The trading rules will be applied to incoming data only, and hence do not need past data.