StalkStock

Stock Market fantasy Contest

Software Requirements Specification

v1.0

17 January 2019

Parangjothi C

Lead Software Engineer

Prepared for

IT350—Software Engineering

Instructor: Biju R Mohan, Ph.D.

Winter 2019

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 17.01.19 | Initial SRS upload | PK,PJ,Sanku | Just started SRS |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
| Pj34s | Parangjothi C | Lead Software Engineer | 17.01.19 |
| sanku | Sankarshan Guru | Lead Assistant | 17.01.19 |
| PK | Pratyush Karmakar | Specialist | 17.01.19 |

**Table of Contents**

**REVISION HISTORY I**

**DOCUMENT APPROVAL II**

**1. INTRODUCTION 1**

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

**2. GENERAL DESCRIPTION 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

**3. SPECIFIC REQUIREMENTS 2**

3.1 External Interface Requirements 3

*3.1.1 User Interfaces 3*

*3.1.2 Hardware Interfaces 3*

*3.1.3 Software Interfaces 3*

*3.1.4 Communications Interfaces 3*

3.2 Functional Requirements 3

*3.2.1 <Functional Requirement or Feature #1> 3*

*3.2.2 <Functional Requirement or Feature #2> 3*

3.3 Use Cases 3

*3.3.1 Use Case #1 3*

*3.3.2 Use Case #2 3*

3.4 Classes / Objects 3

*3.4.1 <Class / Object #1> 3*

*3.4.2 <Class / Object #2> 3*

3.5 Non-Functional Requirements 4

*3.5.1 Performance 4*

*3.5.2 Reliability 4*

*3.5.3 Availability 4*

*3.5.4 Security 4*

*3.5.5 Maintainability 4*

*3.5.6 Portability 4*

3.6 Inverse Requirements 4

3.7 Design Constraints 4

3.8 Logical Database Requirements 4

3.9 Other Requirements 4

**4. ANALYSIS MODELS 4**

4.1 Sequence Diagrams 5

4.3 Data Flow Diagrams (DFD) 5

4.2 State-Transition Diagrams (STD) 5

**5. CHANGE MANAGEMENT PROCESS 5**

**A. APPENDICES 5**

A.1 Appendix 1 5

A.2 Appendix 2 5

# **1. Introduction**

## **1.1 Purpose**

The purpose of this document is to present a detailed description of the Stock Market Fantasy Contest. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stockholders and the buyers of the system.*.*

## **1.2 Scope**

This software system will be a Stock Market Fantasy Contest for a stockholders and buyers of the stocks. This system will be designed to implement a mock stock market to encourage investors to improve their stock investing skills. More specifically, this system is designed to attract students and investors and entrepreneurs with little or no knowledge to learn more about stocks and better invest in the future in real time stocks. The software will facilitate communication between stockholders and the buyers. The system also contains a relational database containing a list of available stocks and the buyers and sellers of the stocks.

**1.3 Definitions, Acronyms, and Abbreviations**

*This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.*

## **1.4 References**

*This subsection should:*

*(1) Provide a complete list of all documents referenced elsewhere in the SRS, or in a separate, specified document.*

*(2) Identify each document by title, report number - if applicable - date, and publishing organization.*

*(3) Specify the sources from which the references can be obtained.*

*This information may be provided by reference to an appendix or to another document.*

## **1.5 Overview**

*The next chapter, the Overall Description section, of this document gives an*

*overview of the functionality of the product. It describes the informal requirements and is*

*used to establish a context for the technical requirements specification in the next chapter.*

*The third chapter, Requirements Specification section, of this document is written*

*primarily for the developers and describes in technical terms the details of the*

*functionality of the product.*

*Both sections of the document describe the same software product in its entirety,*

*but are intended for different audiences and thus use different language.*

# **2. General Description**

## **2.1 Product Perspective**

A **stock market**, **equity market** or **share market** is the aggregation of buyers and sellers (a loose network of economic transactions, not a physical facility or discrete entity) of stocks (also called shares), which represent ownership claims on businesses; these may include securities listed on a public stock exchange, as well as stock that is only traded privately. Examples of the latter include shares of private companies which are sold to investors through equity crowdfunding platforms. Stock exchanges list shares of common equity as well as other security types, e.g. corporate bonds and convertible bonds.

StalkStock aims to portray the underlying concept of exchange through an efficient implementation of the current marketplace*.* The main goal is to provide a platform for investors and brokers for exchange of goods. The platform will serve as a hub to a variety of market trades facilitating to every need of the users.

StalkStock derives its data from existing Stock Market APIs. This data would be used to provide a complete description of the market at the current time to facilitate pricing and efficient exchanges.

## **2.2 Product Functions**

StalkStock is a web based applications which will have 3 main interfaces

1. A Login interface : A login page for the user of the application.
2. An Exchange interface : An interface for users to buy or sell stocks.
3. A Result interface : An interface where users can check their progress.

## **2.3 User Characteristics**

The Buyers or Investors is expected to be Internet literate and be able to buy stocks in the mock stock market.

The Sellers are expected to be Internet literate and to be able to update and display their stocks on the mock stock market.

**2.4 General Constraints**

Gephi is developed in Java, it uses OpenGL for its visualization engine and has been built on top of the NetBeans Platform. It uses a modular design where every feature is wrapped into a separate module and the modules depend on each other through well-written APIs. There are several APIs available to make plugin development easy.

## **2.5 Assumptions and Dependencies**

The application is web based. Therefore the user is expected to have a browser with flash.

# **3. Specific Requirements**

*This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project’s software design, implementation, and testing.*

*Each requirement in this section should be:*

* *Correct*
* *Traceable (both forward and backward to prior/future artifacts)*
* *Unambiguous*
* *Verifiable (i.e., testable)*
* *Prioritized (with respect to importance and/or stability)*
* *Complete*
* *Consistent*
* *Uniquely identifiable (usually via numbering like 3.4.5.6)*

*Attention should be paid to the carefully organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.*

## **3.1 External Interface Requirements**

### **3.1.1 User Interfaces**

### **3.1.2 Hardware Interfaces**

### **3.1.3 Software Interfaces**

### **3.1.4 Communications Interfaces**

## **3.2 Functional Requirements**

*This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.*

### **3.2.1 <Functional Requirement or Feature #1>**

3.2.1.1 Introduction

3.2.1.2 Inputs

3.2.1.3 Processing

3.2.1.4 Outputs

3.2.1.5 Error Handling

### **3.2.2 <Functional Requirement or Feature #2>**

…

## **3.3 Use Cases**

### **3.3.1 Use Case #1**

### **3.3.2 Use Case #2**

…

## **3.4 Classes / Objects**

### **3.4.1 <Class / Object #1>**

3.4.1.1 Attributes

3.4.1.2 Functions

<Reference to functional requirements and/or use cases>

### **3.4.2 <Class / Object #2>**

…

## **3.5 Non-Functional Requirements**

*Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).*

### **3.5.1 Performance**

### **3.5.2 Reliability**

### **3.5.3 Availability**

### **3.5.4 Security**

### **3.5.5 Maintainability**

### **3.5.6 Portability**

## **3.6 Inverse Requirements**

*State any \*useful\* inverse requirements.*

## **3.7 Design Constraints**

*Specify design constraints imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.*

## **3.8 Logical Database Requirements**

*Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.*

## **3.9 Other Requirements**

*Catchall section for any additional requirements.*

# **4. Analysis Models**

*List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.*

## **4.1 Sequence Diagrams**

## **4.3 Data Flow Diagrams (DFD)**

## **4.2 State-Transition Diagrams (STD)**

# **5. Change Management Process**

*Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.*

# **A. Appendices**

*Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.*

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## **A.1 Appendix 1**

## **A.2 Appendix 2**