**Software Requirements**

**Specification**

**for**

**CSCI222 Assignment 1**

**Version 1.0 approved**

**Prepared by <author>**

**<organization> <date created>**

**Copyright © 2002 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.**

**Software Requirements Specification for <Project> Page ii**

# Table of Contents

[Table of Contents 2](#_Toc474250242)

[Revision History 2](#_Toc474250243)

[1. Introduction 1](#_Toc474250244)

[1.1 Purpose 1](#_Toc474250245)

[1.2 Document Conventions 1](#_Toc474250246)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc474250247)

[1.4 Project Scope 1](#_Toc474250248)

[1.5 References 1](#_Toc474250249)

[2. Overall Description 2](#_Toc474250250)

[2.1 Product Perspective 2](#_Toc474250251)

[2.2 Product Features 2](#_Toc474250252)

[2.3 User Classes and Characteristics 2](#_Toc474250253)

[2.4 Operating Environment 2](#_Toc474250254)

[2.5 Design and Implementation Constraints 2](#_Toc474250255)

[2.6 User Documentation 3](#_Toc474250256)

[2.7 Assumptions and Dependencies 3](#_Toc474250257)

[3. System Features 3](#_Toc474250258)

[3.1 System Feature 1 3](#_Toc474250259)

[3.2 System Feature 2 (and so on) 4](#_Toc474250260)

[4. External Interface Requirements 4](#_Toc474250261)

[4.1 User Interfaces 4](#_Toc474250262)

[4.2 Hardware Interfaces 4](#_Toc474250263)

[4.3 Software Interfaces 4](#_Toc474250264)

[4.4 Communications Interfaces 4](#_Toc474250265)

[5. Other Nonfunctional Requirements 5](#_Toc474250266)

[5.1 Performance Requirements 5](#_Toc474250267)

[5.2 Safety Requirements 5](#_Toc474250268)

[5.3 Security Requirements 5](#_Toc474250269)

[5.4 Software Quality Attributes 5](#_Toc474250270)

[6. Other Requirements 5](#_Toc474250271)

[Appendix A: Glossary 6](#_Toc474250272)

[Appendix B: Analysis Models 6](#_Toc474250273)

[Appendix C: Issues List 6](#_Toc474250274)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# 1. Introduction

## 1.1 Purpose

This SRS describes the software functional and non-functional requirement for release version 1.0 of the Warehouse Management System.

The purpose of the Warehouse Management System is to provide efficient bookkeeping of stock in a Warehouse environment. The system has a login feature that allows the personnel with the corresponding privileges to login in to the system while safely securing the data stored in the system. In the case where there are 3 unsuccessful login attempts, the user will be locked out of the system. Someone who is of an admin privilege would have to login to the system and change the user locked out status and change his password so that the user would be able to login.

Another purpose of the software, is the ability to add/update records, edit records, searching and display of stock based on price range and quantity. There are four options the stock can be displayed, daily summary, weekly summary, monthly summary and yearly summary. The software can also add transactions, remove transactions and search transactions.

## 1.2 Document Conventions

The format of this SRS is simple. The titles and content headers are bold face and indentations are used for general topics. The whole document is in Times New Roman.

## 1.3 Intended Audience and Reading Suggestions

This document is intended for the system developers, project managers, testers and lastly the clients. The document will be sent to the client to seek their opinion and validation on anything else they wish to add to the system. Once they give their approval, the production of the system will start.

The suggested reading sequence for reading the document would be from top down Part 1.0 to Part 6.0.

## 1.4 Project Scope

This software is designed for a client who owns a warehouse and would like to have an efficient way of recording and retrieving stock information.

This document will help in creating an effective warehouse tool to streamline process and gives a better understanding of day to day operation on how a warehouse is performing which would in turn help the users make critical management decision based on the data input in the system .While still remaining user friendly and less training would be needed to learn how to use the Warehouse management tool.

## 1.5 References

There are no references for this document at the moment.

# 2. Overall Description

## 2.1 Product Perspective

The application is a new application and is self-contained, with no dependencies to external resources. The application will utilize text files for storage and will not require any form of third party support.

## 2.2 Product Features

The product’s purpose is to facilitate management of stock and it should be able to facilitate log in and checking of user role. A user will be locked out if an incorrect password is provided for more than three times. A user should be able to add, modify and remove stock elements to the system, search for stock elements with various search options and choose from different summary options. An admin should be able to add, modify and remove users from the system.

## 2.3 User Classes and Characteristics

There will be two user classes, Admin which facilitates with administration, creation and modification of user accounts and User; which utilizes the application to perform its core functionalities.

## 2.4 Operating Environment

The application will be written in C++ and will be implemented in Linux/Unix Systems

## 2.5 Design and Implementation Constraints

The application will be written in C++ and is to be implemented on Linux/Unix Systems, the system will display information in a simple textual “menu-select” style of user interface and will read and write to 2 separate text files. One text file will consist of usernames and passwords, sensitive information will have to be encrypted. Multiple accessing and modification of the text files might lead to corruption of data

## 2.6 User Documentation

The user documentation will be included in this SRS (4)

## 2.7 Assumptions and Dependencies

Assumes that the platform used will be a Linux/Unix based system

**Software Features**

1. **Updating stocks**

**Description and priority**

The users will be able to record the stocks that are outgoing or incoming easily. This feature is of high priority as it’s important to know the number of stocks that are currently in store.

**Stimulus/Response sequences**

After logging in, the user will be capable of updating the stock through the use of updateStock() function in ListOfStock.

**Functional requirements**

Req – 1: To be able to record stocks only after successful login.

Req – 2: Error messages should be shown if negative numbers are inputted.

1. **Stocks tracking**

**Description and priority**

Managers will be able to track the stocks by their category and/or sub category. Medium priority as it is not as important as other functions.

**Stimulus/Response sequences**

Upon tracking, the list of stocks will be shown to the user.

**Functional requirements**

Req - 1 : Correct information should be shown upon request.

1. **Sorting**

**Description and priority**

Users can search for either specific stocks in their respective categories or display a list of stocks according to their price range as well as the quantity in either ascending or descending order. Medium/low priority as it only gives readability and does not provide anything substantial.

**Stimulus/Response sequences**

ListOfStock class is accessed through ApplicationController, and you are able to sort according to descending or ascending through the aptly named sortInDescending and sortInAscending functions.

**Functional requirements**

Req – 1 : Sorting functions must be done correctly, checking the second letter if the first letters matches, and so on.

1. **Summary report**

**Description and priority**

There will be a summary report daily, weekly or monthly that will show the total incoming and outgoing stock details. Low priority as it simply makes it easier for the user, but it is of low importance.

**Stimulus/Response sequences**

Summary reports are accessed through the ApplicationControllers’ stockSummaryReport function.

**Functional requirements**

Req – 1 : Total profits, stocks sold, stocks remaining and so on should be correctly displayed upon the functions’ activation.

1. **Login**

**Description and priority**

There will be a login feature where you are able to encrypt your data if you desire. Furthermore, if there are three consecutive unsuccessful login attempts, the record will be marked as locked and the user will be barred from logging into the system. This is of high priority as login is important to ensure that only the authorized personnel are able to enter the system.

**Stimulus/Response sequences**

The user is first presented with a login screen. Upon a successful login by typing in the correct username and password through User class, loginStatus variable in User class will be changed and the user will be allowed to access the program. If the user fails to enter the correct login information three times, the account will be locked through the use of lockUserAcc variable in ApplicationController.

**Functional requirements**

Req – 1 : Access should be denied if the username or password is inputted incorrectly.

Req – 2 : User should be barred from retrying after failing three times in a row.

Req – 3 : Users should NOT be allowed entry upon the input of an invalid username or password.

# 

# 4. External Interface Requirements

## 4.1 User Interfaces

The program will have the interactive menu for the users to access the functions of the program. They will have to log in first before accessing and selecting the user type to log in. After logging in, the user can select the functions from the options shown. If the user attempts to log in unsuccessfully for three times, a pop out dialogue appears and inform the user that his account is locked. Therefore, the program has a management system for the user accounts for situation where an account is locked. The administrator will need to log to the system to unlock the respective account. There will be an option in the menu for the user to log out to exit the program.

## 4.2 Hardware Interfaces

No interaction with other hardware interfaces except the computer itself.

## 4.3 Software Interfaces

The program stores its’ data in a text file. The software does not interact with any other software.

## 4.4 Communications Interfaces

No communication interfaces are needed as the program runs locally without accessing the internet.

# 5. Other Non-functional Requirements

## 5.1 Performance Requirements

The system must response within 5 seconds when navigating through the menu, displaying output and login.

## 5.2 Safety Requirements

The stock reaches below a certain threshold for quantity available, a warning message will be display.

## 5.3 Security Requirements

One account could only be login at a time and the access rights is based on the account types.

After 3 failed login attempts, the account will be locked.

## 5.4 Software Quality Attributes

The system is intuitive and easy for the user to navigate around. Clicking on the options will result in different response times with a few seconds of deviation. User should be well equipped to use the program with an hour of training or reading up. The user should make less than 5 errors a week to be considered well equipped.

The password for the login needs to be strong like having an alphanumeric combination as password.

# 6. Other Requirements

|  |  |
| --- | --- |
| Constraint | Description |
| 1 | Have to be implemented in c++ |
| 2 | Must run on Linux operating system. |
| 3 | Menu select style interface |

# 

# Appendix A: Glossary

SRS – Software requirement specification

# Appendix B: Analysis Models

# Appendix C: Issues List