|  |
| --- |
| Team C1 |
| Software Requirements Specification for CSCI222 Assignment 2 |
| Version 1.0 |

|  |
| --- |
| Deston, Eugene, Darwin, Joshua, Benedict, Randy  2-18-2018 |

**Table of Contents**

[1. Introduction 2](#_Toc506723270)

[1.1. Purpose 2](#_Toc506723271)

[1.2. Document Conventions 2](#_Toc506723272)

[1.3. Intended Audience and Reading Suggestions 2](#_Toc506723273)

[1.4. Project Scope 2](#_Toc506723274)

[1.5. References 3](#_Toc506723275)

[2. Overall Description 4](#_Toc506723276)

[3. System Features 5](#_Toc506723277)

[3.1. Login function 5](#_Toc506723278)

[3.1.1. Description and Priority 5](#_Toc506723279)

[3.1.2. Stimulus/Response Sequences 5](#_Toc506723280)

[3.1.3. Functional Requirements 5](#_Toc506723281)

[3.2. Staff Control Panel 5](#_Toc506723282)

[3.2.1. Description and Priority 5](#_Toc506723283)

[3.2.2. Stimulus/Response Sequences 5](#_Toc506723284)

[3.2.3. Functional Requirements 5](#_Toc506723285)

[3.3. Admin Control Panel 5](#_Toc506723286)

[3.3.1. Description and Priority 5](#_Toc506723287)

[3.3.2. Stimulus/Response Sequences 5](#_Toc506723288)

[3.3.3. Functional Requirements 5](#_Toc506723289)

# Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the functionalities of the Warehouse Management Tool system. This document will cover each of the system’s intended features, as well as offer a preliminary glimpse of the software application’s User Interface (UI). The document will also cover hardware, software, and various other technical dependencies.

## Document Conventions

This document is unique. Its scope is to describe the requirements of the Warehouse Management Tool system and the interfaces for it. This document has been created before the actual program is constructed, hence is it important to note that future changes to the program should be included in this requirement specification document, in order to maintain its usefulness and relevance.

## Intended Audience and Reading Suggestions

This document may be used by the staff or clients of the Warehouse Management Tool system. With our user-friendly interface, users can easily understand the features and the operations of this system. They can also be informed of all the possibilities the program provides. Users can get a general image of the program and use it with better ease and understanding. This document should be studied by the programmers or system admins, to better understand the program. With a better understanding of the program, the chances to identify and prevent possible problems increases. It also allows for expandability to the program.

## Project Scope

The project scope is to develop a user-friendly interface to perform a simple stock checking and inventory management. The UI will be the terminal in a windows or Linux operating system. An effective stock checking and inventory management tool is essential to streamline and optimize processes, long range planning to get the best efficiency. An in-depth summary report also provides users with a better understanding of the performance of the operation. The management teams are able to take the data provided to make critical management decisions that will impact the organization.

It is critical for the warehouse management to have an effective system in place due to various reasons such as security, accuracy, availability and so on. We are requested to use a test-driven approach to develop the warehouse management tool which needs to meet the following requirements.

List of features include:

* Ability to add/update records on the incoming stock and outgoing stock easily.
* Track the stock in accordance to their category and sub-category.
* Ability to search and display available stock for a particular category and/or sub-category.
* Ability to search and display stock in accordance to price range and quantity in ascending and descending order.
* Daily, weekly or monthly summary report of total incoming and outgoing stock details.
* Provide login authentication process.
* Login security feature with option of encrypting the data stored.
* If there are three unsuccessful login attempts, the record will be marked as “locked” and one is not allowed to login to the system.
* The system shall encrypt and decrypt the password.
* Provide stock item alerts when it drops below a threshold set previously.

## References

CSCI 222 Assignment 2 Report template

# Overall Description

**2.1** **Product Perspective**

This product is a new, self-contained product that will be designed and programmed from scratch.

**2.2 Product Features**

This product will feature several functionalities, admin users will be allowed to create, delete and reset user accounts. The staff account will only be able to manage the inventory and stocks and print a summary report in the following formats: daily, weekly, monthly and yearly.

**2.3 User Classes and Characteristics**

This product will have two major user classes, mainly the controller and stock class. The frequency of usage will be high as once the staff user logs in, the controller and stock classes will be used.

**2.4 Operating Environment**

This program will run in either windows or Linux based operating system.

**2.5 Design and Implementation Constraints**

Given the very little time allowed for this assignment, our team has decided on C++ and the base environment would be Ubuntu to kickstart off this product.

**2.6 User Documentation**

Not applicable as this is still in the design phase.

**2.7 Assumptions and Dependencies**

We assumed that this program will be running in windows or Linux based operating system as user has requested for program to be coded in C++.

We also assumed that the system admin would have some C++ background and Linux background in order to provide user training for all staff members and maintain staff accounts.

# System Features

## Login function

### Description and Priority

Allows the user to log in then proceeds to authenticate the user before directing them to the appropriate control panel based on their role. High priority.

### Stimulus/Response Sequences

* Login
  + Wrong username/ password
  + Account locked out
* Role Validation
  + Staff
  + Admin

### Functional Requirements

* REQ-1: Login function must be able to authenticate user based on entered username and password.
* REQ-2: Login function must be able to encrypt password when authenticating user.
* REQ-3: Login function must be able to read the user’s roles and restrict access to certain functions.
* REQ-4: Users with admin roles may not be locked out of their accounts.

## Staff Control Panel

### Description and Priority

Used primarily by staff to add and remove stocks from database. They can also display the summary report of a period of time. High priority.

### Stimulus/Response Sequences

* Search Product
* Edit Product
* Add Product
* Remove Product
* Summary Report
* Retrieve Product

### Functional Requirements

* REQ-1: Must be able to encrypt and decrypt the database file when manipulating fields.

## Admin Control Panel

### Description and Priority

Used by staff with admin roles to manage accounts. Includes the ability to Medium Priority.

### Stimulus/Response Sequences

* Create account
* Reset Account
* Delete Account
* View Account

### Functional Requirements

* REQ-1: Must only be accessible by users with admin roles
* **4.** **External Interface Requirements**
* 4.1 User Interfaces

Upon start-up of program a simple GUI will appear. First is the login interfaces basically user just need to enter username and password. After successful login there will be a main menu consisting of what the user is able to do within the program itself. Any error input message will be for like example “error, invalid input”

* 4.2 Hardware Interfaces

As the program can be run in either Linux or windows based operating systems and is not a complicated program all user need is a working laptop or desktop and the program will run fine.

* 4.3 Software Interfaces

For this program there will not be any user of integrated commercial components, as it will be done from scratch and it does not need internet connection for program to work and all data will be store in a text file so no data or message will be coming into the system or leaving.

* 4.4 Communications Interfaces

No communications interfaces is required as the program does not need to have internet connection for it to run is a standalone program.

**5. Other Non-functional Requirements**

* 5.1 Performance Requirements

The program has been tested to perform at optimum peak performance to avoid backlogs. The program has also been stress tested to ensure that the program would not break down and potential ruin the user’s desktop, thus resulting in further damage.

* 5.2 Safety Requirements

The program would not comprise of any machinery and would only require a desktop/laptop to operate. Therefore, there are minimum safety risks.

* 5.3 Security Requirements

The program hashes the user’s password in SHA256 and appends an additional SALT to provide maximum security with little performance downfall. The added security function significantly reduces the chances of any cyber-attacks being successfully.

* 5.4 Software Quality Attributes

The program is relatively simple to maintain as it has been put through test cases to ensure that no unwanted features can occur.

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.> Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>